Building The Big Walker - Mechanical Design

SERVO MAGAZINE

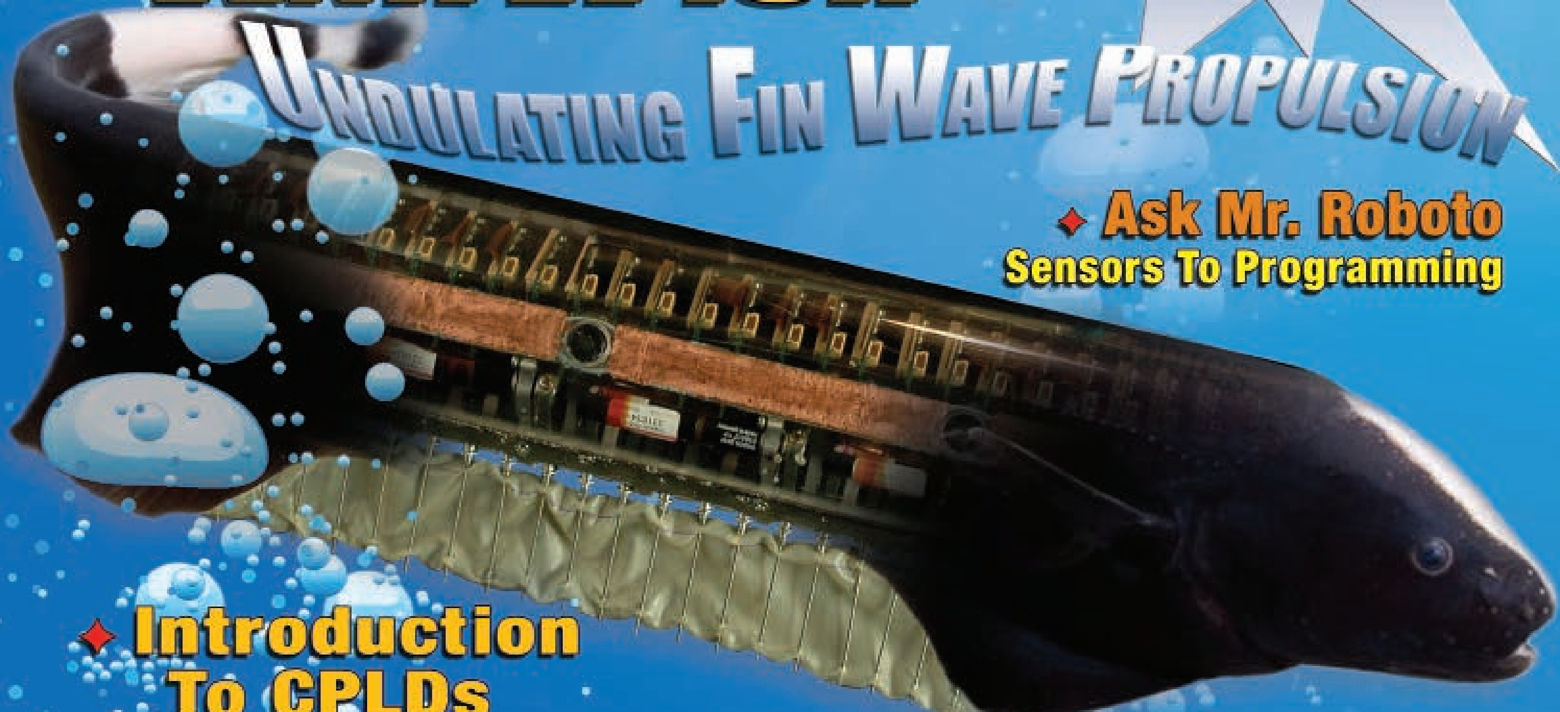
FOR THE ROBOT INNOVATOR

www.servomagazine.com

March 2011

Mimicking The Black Ghost KnifeFish

Undulating Fin Wave Propulsion



Get Ready RoboGames Is Coming!

- Ask Mr. Roboto Sensors To Programming
- Introduction To CPLDs
- Energy Harvesting
- Telepresence

U.S. $5.50 CANADA $7.00

03

0 71486 02422 4

INVEST in the BEST

YOUR ROBOT WILL THANK YOU

At Hitec, we know how much time and money you dedicate to your robot hobby. So why not make sure your servos are delivering what they promise? Our technologically advanced mega servos are tough enough for even your most challenging projects. Designed with our newest high resolution "G2.5" 12-bit programmable digital circuit and indestructible titanium gears, the HS-7980TH and HS-M7990TH give mega torque and speed with pinpoint accuracy. Built to last, these servos bring unprecedented power and sustainability to your investment.

Our HS-7980TH employs durable mechanical potentiometer technology while the HS-M7990TH utilizes the first ever magnetic encoder.

	6 Volts		7.4 Volts				
Model	Speed	Torque	Speed	Torque	Part#	Dimensions	Weight
HS-7980TH	0.20	500 oz.in	0.17	611 oz.in	37980S	1.72 x 0.88 x 1.57 in	2.70 oz
HS-M7990TH	0.20	500 oz.in	0.17	611 oz.in	37990S	1.72 x 0.88 x 1.57 in	2.70 oz

HITEC

We Serve you Right!

12115 Paine Street, Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com


METAL GEARMOTORS
SERVOS
FORCE SENSOR?
METAL GEARMOTOR
MOTOR CONTROLLER
SERVO CONTROLLER
ORDER AT POLOLU.COM
NEEDED:
LASER-CUT PIECES
SHARP SEN
SERVOS
JRK CONTROL
CHECK OUT SMC:
USE COUPON CODE!
SERVO8556
Take your design from
idea to reality
Pololu
Engage Your Brain
www.pololu.com

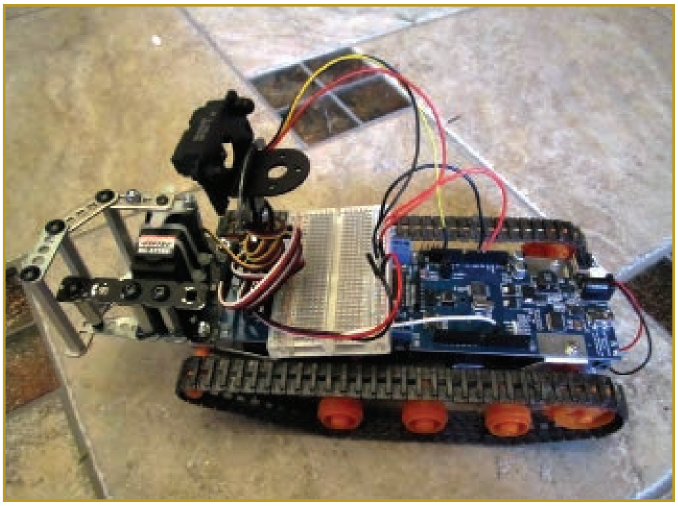
SERVO MAGAZINE

03.2011

VOL. 9 NO. 3

Did you know that each article in *SERVO Magazine* has its own webpage? It's where you go for downloads, comments, updates, corrections, or to link to the article in the digital issue. The unique link for each webpage is included with the article.

You can also visit article pages from back issues at **www.servomagazine.com**. Just select the ***Past Issues*** tab from the ***About SERVO*** drop down menu, click the Table of Contents link, and the article name.



PAGE 70

Departments

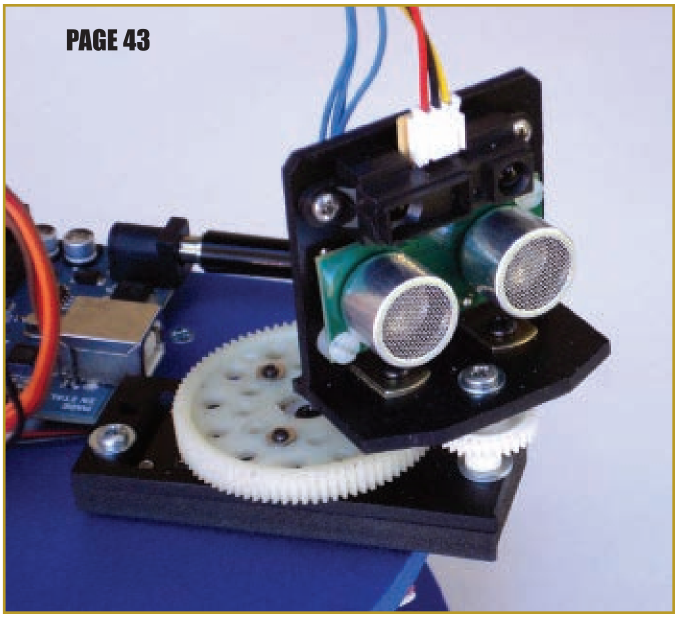
Columns

PAGE 08

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...

PAGE 43



PAGE 56

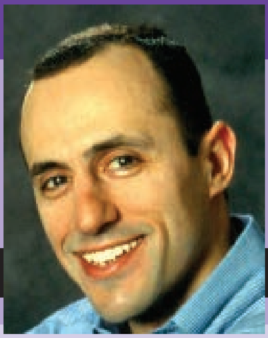
The Combat Zone...

Features

Events

Mind / Iron

by Bryan Bergeron, Editor



Building with Repurposing in Mind

In these cost-conscious times, the economics of robotic experimentation may be at odds with your budget. A DIY robot with microprocessor, a few sensors, and two or more servos can easily run $100 or more in parts. And once you've built and tested your robot, you'll inevitably want to upgrade or significantly improve the design. That's when it starts to get expensive.

You can minimize your total cash outlay by building your robots with repurposing in mind. That is, assume your robotic creations are temporary, and that one day soon you'll be using all or some of the parts in some other robotics project. Tearing down one of your robots needn't be an arduous, lengthy task if you go about your prototyping with repurposing in mind.

One obvious approach is to use a solderless breadboard for your work. The larger boards with aluminum backing and multiple binding posts for power can be great for basic circuit prototyping, before you deploy the circuit on a mobile platform. However, except for small boards, breadboards are often too cumbersome and heavy for deploying on a robotic platform. The solderless breadboards that are integrated with the robots from Parallax (**www.Parallax.com**) and the mini 1.8"x1.4"self-adhesive breadboards from SparkFun (**www.sparkfun.com**) are useful, but limited in component capacity. When you need to work with more than a dozen components or with unwieldy components that don't insert neatly into a solderless breadboard, you'll have to turn to other options.

You can avoid soldering ICs directly into a circuit and use IC sockets instead. I don't use the gold-plated variety, but whatever is on sale from suppliers like All Electronics (**www.allelectronics.com**). At 15–20 cents, sockets are a no-brainer when it comes to repurposing ICs. You'll spend that much on solder-wick if you try to unsolder an IC from a board, and you may damage the IC in the process.

One step up from IC sockets is to use breakout boards, or miniature circuit board modules with connections to all or most of the pins on an IC or other device. I like to use breakout boards from SparkFun, but instead of soldering wire to the boards directly, I add headers so that I can use cables with connectors. That way, I can move the modules from project to project in minutes. This module approach is especially useful for power supplies.

Many of the sensor breakout boards from Parallax come with headers designed for easy insertion into breakout boards. These work fine as long as you can fit your creation in a breadboard. However, the vertical pins can be difficult to work with cables and connectors, unless you can mount the board upside down.

Another way to avoid soldering directly to components is to use wire-wrapping to connect components, sockets, and pins. You'll need to invest in a wire-wrap tool ($6 at RadioShack) and 30-gauge wire-wrapping wire ($9 for a 100' spool; SparkFun). The great thing about wire-wrapping is that once you get the hang of it, it's much faster than soldering. And you can quickly and harmlessly unwrap wire from a pin or component. There's no soldering iron, fumes, or potential solder splashes to deal with – all plusses for experimenters of any age.

The downside of wire-wrapping is that the tool and special wire can be expensive – be careful where you shop. The same tool available from RadioShack for $6 sells for $16 to $56 on some websites. Another issue is that 30-gauge wire has limited current carrying capacity. You'll want to consider heavier gauge wire and solder or crimp-on lugs, and other substantial connectors for battery and motor connections.

When possible, use terminal blocks instead of soldering wire to expensive circuit boards. For example, the screw shield for the Arduino — available from The RobotShop (**www.robotshop.com**) and others — is a handy prototyping tool if you work with the Arduino. In fact, shields in general are a great way to repurpose your microprocessor boards. Simply unplug the shield and move your processors to your next project.

My last suggestion is to document what you build. Sketch a schematic of your device before you retire it. With schematic in hand, you won't have to examine your collection of robots to find the one with the 3V power supply, or the one in which you used the 16 MHz version of a processor.

At some point in your prototyping, you may develop a 'keeper' – something you'll never disassemble for parts. When you reach that stage, a custom printed circuit board and soldered components will usually provide the best reliability and smallest footprint. Good luck with your projects. **SV**

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Fred Eady, Kevin Berry, Bryce Woolley, Evan Woolley, Gregory Intermaggio, Gordon McComb, David Ward, Daniel Albert, Pete Smith, James Baker

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

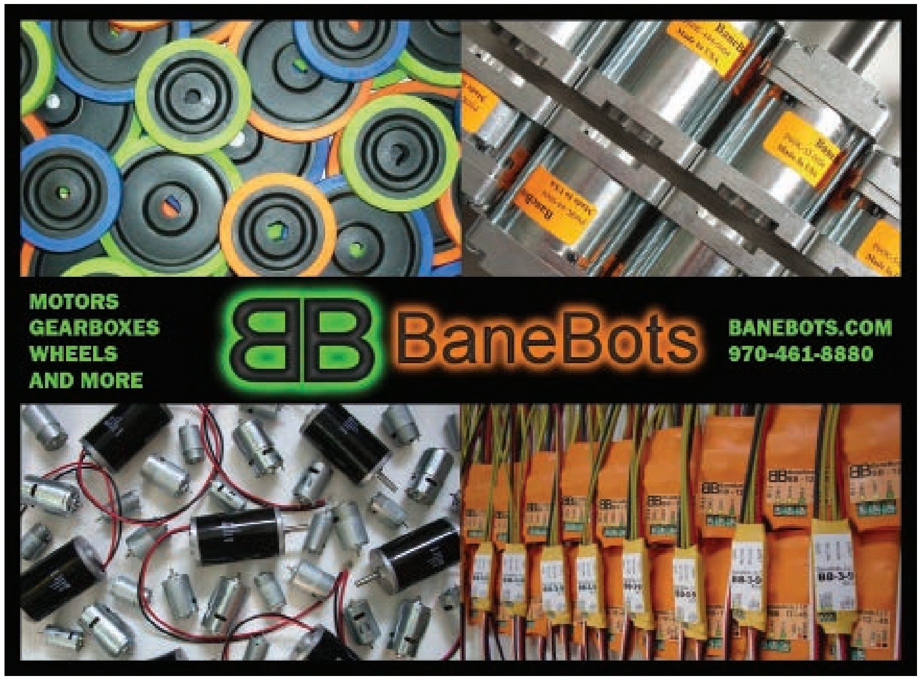
ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

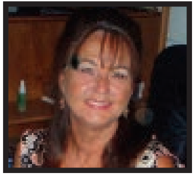
PRODUCTION/GRAPHICS
Shannon Christensen

Copyright 2011 by
T & L Publications, Inc.
All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

Printed in the USA on SFI & FSC stock.



Robytes

by Jeff and Jenn Eckert

Koreans Learn English from Eggbots

Late last year, a fleet of 29 robots was unleashed by the Korea Institute of Science and Technology (KIST, **www.kist.re.kr**) in a pilot program designed to teach English to youngsters in Daegu city schools. The little egg-shaped "Engkey" units stand about 1 m (3.3 ft) high and sport TV facial displays and arms that flap like wings. The facial avatar shown is that of a Caucasian woman, and it speaks to the students, reads to them, and so forth. The bot even provides music and dances with the students. The units are actually remotely controlled by teachers who can see and hear the students via the control system. Their facial expressions are monitored by cameras, and these expressions are automatically reflected on the avatars' faces, thereby displaying amusement, satisfaction, annoyance, death threats, etc. According to an official in the education office, "The kids seemed to love it since the robots look, well, cute and interesting." The accompanying photo doesn't quite support that assessment, but we'll accept it at face value. Interestingly, outsourcing seems to be a phenomenon in Korea, as well as elsewhere; the English teachers controlling the Engkey units are based in the Philippines. "Well educated, experienced Filipino teachers are far cheaper than their counterparts elsewhere, including South Korea," it was revealed.


Students in the city of Daegu, slightly engaged with their robotic teacher.

Kid-Controlled Reefbot

Also geared for the kiddies is the Reefbot, recently installed in a two-story tank at the Pittsburgh Zoo and PPG Aquarium (**www.pittsburghzoo.com**). A joint project of the zoo and the Carnegie Mellon Robotics Institute (**www.ri.cmu.edu**), the system is based on a submersible ROV fitted with a video camera, dubbed CLEO (for children learning through education and observation). The 1.5 ft long vessel allows children to navigate throughout the 100,000 gallon saltwater tank, annoy some 30 species of sea life, and even take photos of them. The kids can then compare their images with reference photos to identify the critters. Details are available at **reefbot.com**.


The Reefbot control panel with a child at the helm.

Don't Mess with MAARS

If you're getting tired of the parade of robots that aim to be cuddly and lovable, take a gander at the Modular Advanced Armed Robotic System (MAARS®) developed by QinetiQ North America (**www.qinetiq-na.com**). This menacing little guy aims at something completely different, i.e., enemy forces. MAARS is billed as "the first fully modular ground robot system capable of providing force escalation options for a measured response that fits any situation." Indeed, in the non-lethal configuration, it just broadcasts the operator's voice through loudspeakers and dazzles intruders with an eye-safe laser. The second option is the "less lethal" version which deploys smoke, pepper spray, star clusters, and bean bags to drive away unwelcome visitors. However, if you don't want to coddle the bad guys, you'll need the "lethal" implementation which can be equipped with four grenade launchers and a machine gun that can fire up to 400 rounds of 7.62 cal ammo. Reportedly, three battle bots have been deployed to Iraq but haven't been allowed to fire a shot because of fears that they might go berserk. That seems unlikely, though, as MAARS — which is not fully autonomous — also carries multiple cameras so the operator has a clear view of its surroundings, ensuring that weapons are not inadvertently aimed at civilians, friendly forces, or other inappropriate targets.


The MAARS battle bot in its combat-ready configuration. Courtesy of QinetiQ North America.

Cutting the Pork

Coming out of an election year, we have heard a lot of talk about cutting pork, but the folks at BANSS Germany Gmbh (**www.banss.de**) take it literally. An adaptation of the company's robotic pork processor was recently installed at the Craig Mostyn Group's Linley Valley facility in Wooroloo, Western Australia, at a cost of $700,000. The system uses 3D laser imaging to make precise measurements of each carcass and then programs a robot to do the cutting based on the animal's specific anatomical features, as well as the customer's requirements. Apparently, manual processing often results in stomach breakage which creates a high contamination risk and requires extra handling and trimming. The new system is expected to reduce such occurrences from four to less than one percent which equates to bringing the total number of affected pigs down from 20,000 to no more than 5,000 annually.

3D laser scanning provides exact measurement of pig carcasses for robotic processing. Courtesy of BANSS Germany.

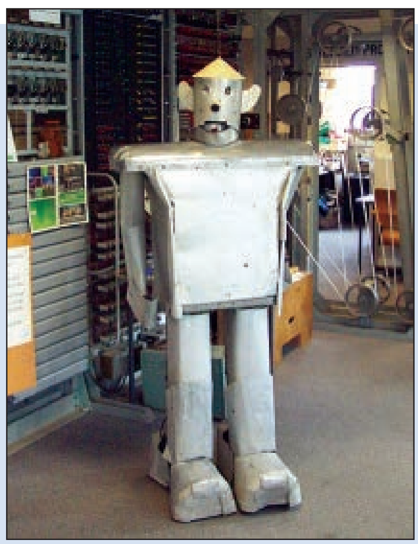
Four-Rotor Flyer Replaces Helicopters

Moving to the realm of bots that are less destructive, we have the R4 inspection robot from — appropriately enough — R4 Robotics (**www.r4robotics.com**). R4 is a Michigan-based startup that has been developing a small flying craft intended for inspection of powerlines and pipelines, windmill blades, towers, and bridges, etc. In addition, it has potential applications in security and law enforcement. A big advantage is that it can eliminate the need for manned helicopters to save on operating costs and eliminate flights in dangerous terrain. The lithium-polymer (LiPo) powered L4 employs four hubcap-size rotors for flight, and it sports a range of sensors including UV, IR, and spectrum-shifted visual (to eliminate haze). It is capable of remote or autonomous operation, and it can fly for about 20 minutes (enough for a five mile round trip) on a charge. L4 can be programmed to remember specific flight paths, and it automatically corrects for wind drift. The unit was originally designed for military use in Germany, but the US company has exclusive rights to build it here.

The R4 flying inspection craft.

Bot Awakens After 45 Years

Finally, if you happen to be passing through the town of Bletchley, England, stop by the National Museum of Computing (**www.tnmoc.org**) and say "hi" to George, who has resided there since late last year. George is a humanoid robot built in 1950 by Tony Sale from the remains of a crashed Wellington bomber, at a cost of about £15. George spent the last 45 years idle in Tony's garage but recently came to life when given an application of oil and two new batteries. He still walks and moves around freely, but the museum reports, sadly, that he can no longer sit down and has lost the use of his photocell eyes, so he can't locate an illuminated bottle of beer as he once could. While you're there, you can also gaze at the rebuilt Colossus, said to be the world's first electronic programmable computer. The Colossus machines were used during WWII to help decipher encrypted Nazi messages. If you can't make it to the UK, you can at least see George in a 1950 newsreel by visiting **www.britishpathe.com/record.php?id=74088**. **SV**



George stands guard over the rebuilt Colossus computer.

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

Knifefish Research Robot Models Undulating Fin Wave Propulsion

By studying the waves the knifefish makes with its undulating, body-length fin, Malcolm A. MacIver, PhD, associate professor and other researchers at Northwestern University plan to learn more about this novel form of fish propulsion so they can develop comparable fins for water-based robots to enhance their mobility, hovering, and station-keeping capabilities.

Robotic fins would make for better propulsion in hovering and station keeping because the current use of propellers offers poor thrust performance at low speeds. "In addition, they are prone to entanglement in weeds and damage due to the need for high spin rates for smooth performance," stated MacIver.

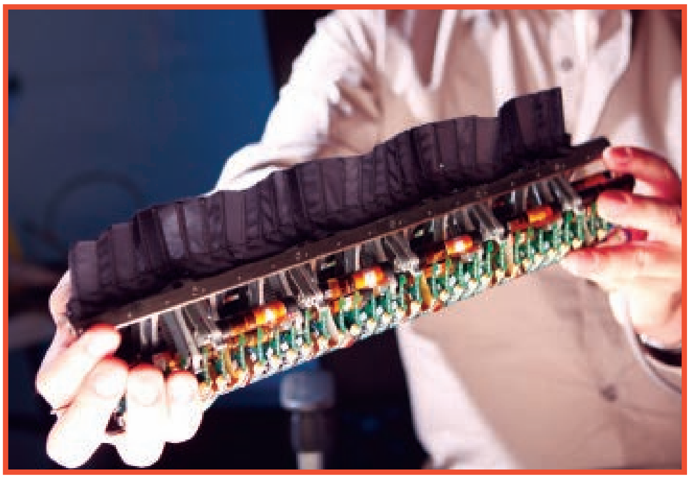
Why and How They Studied the Black Ghost Fish

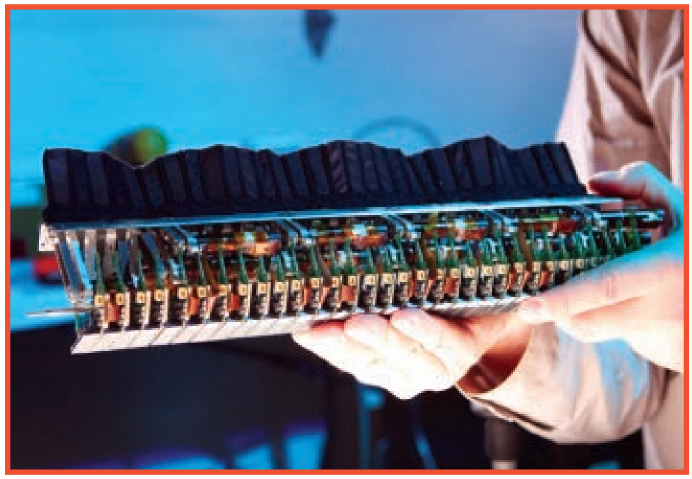
The specific fish under the researcher's thoughtful attention is the black ghost fish — a type of knifefish from South American regions. The fish was selected due to its unique sensing, navigation, and propulsion skills. The fish does object sensing using an electrical discharge it sends out in all directions. The fish can change direction very quickly. When simply moving ahead or back, the fish moves the ribbon fin in waves either front to back or back to front. "The black ghost fish are low speed, high agility specialists. I have shown that they reverse direction in about half a second when swimming at around 10 cm/s," commented MacIver.

The curious navigational wizardry of the black ghost fish occurs when it is hovering or station keeping. The fish can remain in one place and keep its station against the water currents by use of what the researchers call inward counter-propagating waves. The fish creates two recurring, simultaneous waves along its fin. One starts at the front and moves to the tail; the other starts at the tail and moves toward its head; and the two waves meet in the middle. Because the amplitude of each of the two waves tapers to zero when they meet at the middle of the fin, both waves stop at the center of the fin when they meet, according to MacIver.

To demonstrate and study the fish's keen station-keeping capacity, the researchers apply computational fluid dynamics to a computer model of the actual fish while using digital particle image velocimetry (DPIV) to observe and clock the speed of the flow around the robotic fish.

Malcolm MacIver holding his knifefish robot used to research waves created by the real knifefish's body-length underbelly fin. *Photo by Andrew Campbell.*





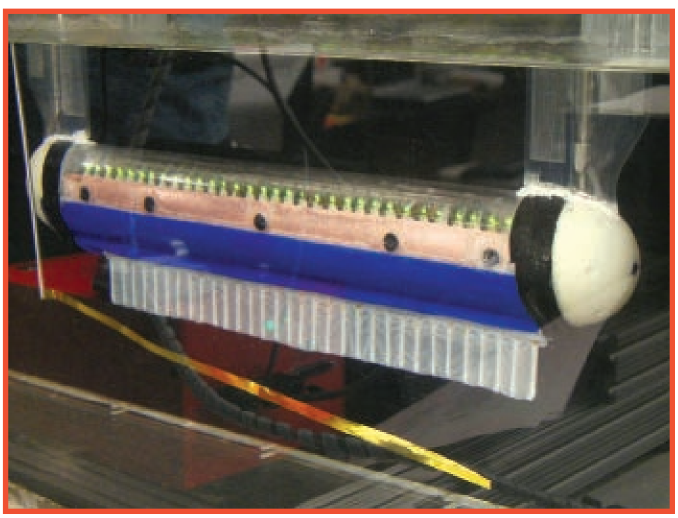
Close-up of knifefish robot and its inner workings. *Photos by Andrew Campbell.*

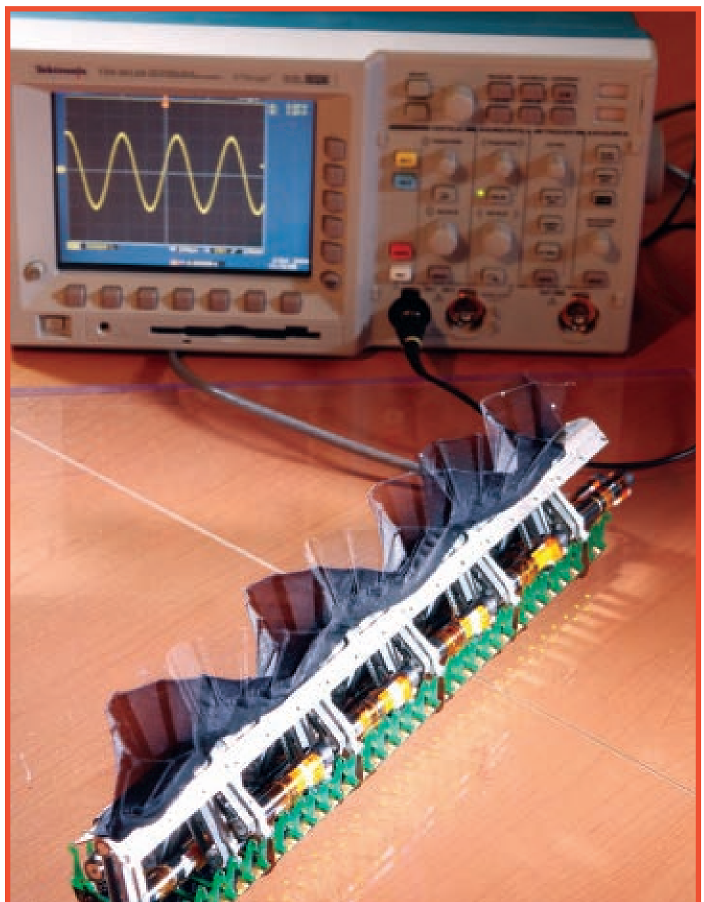
The researchers use the DPIV to see the flow structure of the robot fish in the midsagittal and transverse planes. "If you were facing me, the midsagittal plane is the one that would cut you into two symmetric halves. The transverse plane is a plane at right angles to the spine," defined MacIver. This enables them to reveal the mechanics behind the inward counter-propagating waves of the belly fin to determine how it hovers.

"What DPIV gives us is a way to quantify how the fluid around the fin responds to the prescribed fin motion. So, we are able to say, for example, that inward counter-propagating waves generate a jet that has a peak velocity of 30 cm/s when the fin moves with a traveling wave of 4 Hz, two waves on the fin, with a peak amplitude of 35 degrees. Without DPIV, we have no way of quantifying how the fluid moves, and without that, you cannot fully understand the fluid mechanics of how propulsion from the fin works," MacIver explained.

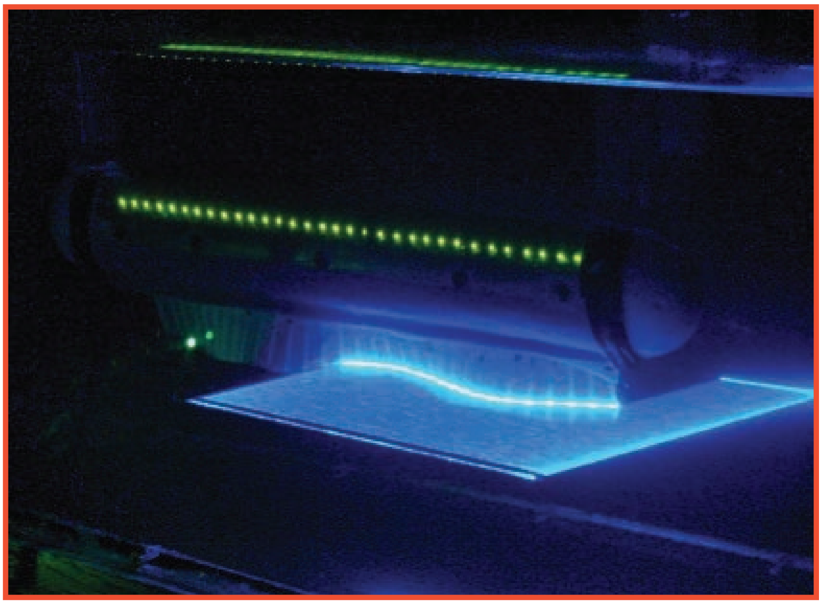
The scientists also studied how these kinds of waves compare with other types such as standing waves, unidirectional waves, and waves starting at the middle of the fin and moving outward toward the head and tail simultaneously. "In a standing wave, the nodes of the waveforms (where the amplitude is zero) do not move. In a unidirectional wave, the nodes are moving, but only in one direction. So, a wave traveling from the front of the fish's fin to the back would be a unidirectional

Knifefish attached to mechanism for submersion in water. *Photo by Malcolm MacIver.*





Knifefish wave at work out of water. *Photo by Andrew Campbell.*



Knifefish operating with blue light effect. *Photo by MacIver.*

Knifefish scale when shown with chopsticks. *Photo by Julio Santos-Munne.*

wave," says MacIver.

Drilling down, the researchers calculated the fin height and length, and the fish's surge, heave, and sway, as well as angular fin deflection and other measurements to create the swimming fish model using fluid dynamics.

The real black ghost fish was observed in a Plexiglas refuge of the type it natively likes to hide in (minus the manmade materials) using a high-speed Casio Exilim camera taking approximately 300 frames per second. Measures of the weight and density of the fish were also taken. The fish model and eventual fish robot started from a cast made of an actual black ghost fish body.

To place the robot fish in the water, and power and control it, the researchers mounted it on a precision ball bearing, linear slide assembly. This is a linear, vertical slide assembly that enables the researchers to measure the upward force generated by the inward counter-propagating waves, explains MacIver. "There is a force sensor on the slide. As the robot generates the inward waves, it pushes against this force sensor, telling us how much force is being generated," commented MacIver.

The robot fish was also observed using two synchronized cameras which were both FastCam Photron APX cameras from Photron. The researchers used these to view the robotic fish from the side and the rear. They calibrated the cameras across the full spatial field of view with DAVIS v. 7.1 software from LaVision of Germany.

"This is software for analyzing the DPIV data. It takes a series of images of the reflective particles, and then runs an algorithm on those to figure out how those particles are moving across the images. Calibration across the full spatial field of view just means that our calibration grid — which is always necessary when first setting up the cameras — filled the entire field of view, so all the data will be properly calibrated," MacIver explained. Images were taken at 500 frames per second for the midsagittal view and 1,000 frames per second for the transverse view in high resolution (1024p x 1024p).

When using the DPIV to observe the flow, the scientists used 50 micrometers in diameter beads for the particles and lit them up with a 10W continuous wave argon ion laser. A mirror was used to deflect the laser light from a point under the water tank up into it.

Mechanics of the Black Ghost Knifefish Robot

The formal name of the robot fish is GhostBot which MacIver and researchers together with a company called Kinea designed and built. "Because we needed 32 independently controlled fin rays in as small a package as possible, the key design element is a stack of 'fish steaks.' Each of these fish steaks is a custom PCB with chips for handling communication and control of a 10 mm RE10 Maxon motor with gear reducers and encoders. They snap together via their communication bus and form an integral structural element," detailed MacIver.

Since each motor is approximately 30 mm in length, it was important for MacIver to come up with a compact way to position them so that the robot did not end up being too long. "We used a CAN bus — frequently used in the automotive industry — for communicating with each of the 32 fish steaks. Outside of the robot, we used a real-time kernal (xPC, from Mathworks) running on a PC104 stack, for generation of control signals. Power and control signals are sent via a multichannel tether. Outside of the Maxon motors and the electrical components on the fish

steaks, all parts of GhostBot were custom made," stated MacIver.

Observations

The black ghost fish hovers by means of creating inward counter-propagating waves along its ventral ribbon fin. This can be used to keep station and provides an upward force to keep the fish — which is slightly denser than water — from sinking. The fish also uses the upward forces to accomplish upward maneuvers.

More specifically, the research team's earlier work demonstrated that the belly ribbon fin creates two colliding central jets which are deflected downward, creating an upward force. The force pushes the fish's body straight up when the two counter propagating jets are creating the same amount of fluid momentum.

The fish can modulate the counter propagating waves to adjust to changes in the ambient water flow and easily modulate its forward and backward movement, even generating upward motion if needed.

Complimenting the fish's fin capabilities is the high density of sensory receptors along the top edge of the fish's body. The majority of the fish's prey are detected directly above the fish's body (owing to this fact).

Conclusion

Northwestern University researchers are in talks with a major US robotics company to begin sharing their technology with them. "There is certainly the interest from a major player to make this a reality. Realistically, it will be a minimum of several years before we see this, if all goes well," explained MacIver.

In the next version of the robot, the researchers will have something that will allow them to test stability and 3D maneuvering of the robot. "In the current version, the robot is really set up for testing 1D (forward and back) movement. We've shown it can do vertical movement through demonstrating that there is more than adequate force for this; but due to instrumentation and mounting constraints, we haven't shown the robot moving vertically," MacIver concluded. **SV**

Resources

MacIver's lab at Northwestern
www.neuromech.northwestern.edu

MacIver's paper on fish motion
www.neuromech.northwestern.edu/publications/MacI01a/MacI01a_prey_capture_behavior_gymnotid_el.pdf

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

This month, there is a mix of questions from sensors to programming. I hope that you learn something new from Mr. Roboto's tower-top lab. Let's dig in!

Q. Mr. Roboto, I have an Arduino Pro 328 using 5V power from SparkFun and a SparkFun FTDI Basic board that I use (over USB) to program it. When I use my Arduino 017 setup, this works fine. But, when I try to program the board using avrdude, it fails saying that "The programmer is not responding." What am I doing wrong?

— David (via email)

A. David, You didn't say how you were invoking avrdude, so I put together a sketch for the Arduino IDE and indeed it worked. I then used avr-gcc to create a simple LED blinker and tried to use avrdude to program my Arduino Pro board (see **Figure 1** for my layout) and sure enough, it failed for me. I started with this invocation to program the board:

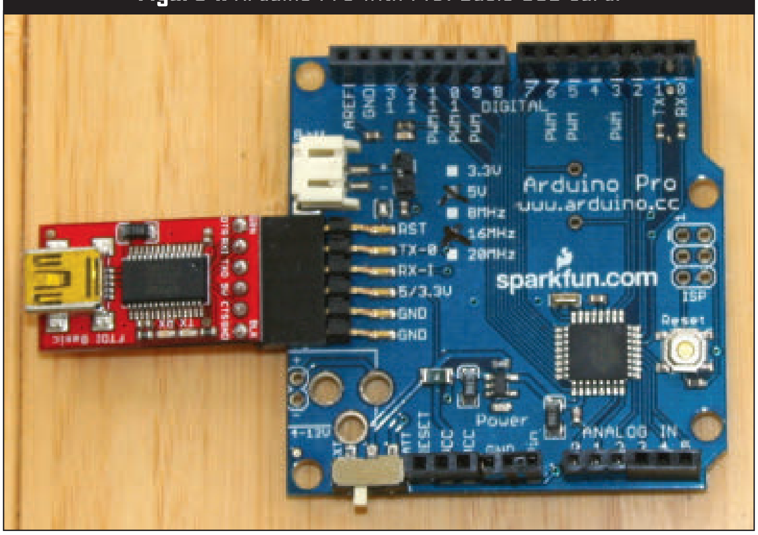


Figure 1. Arduino Pro with FTDI Basic USB card.

```
avrdude -c arduino -p M328p -P
/dev/cu.usbserial-A700e5J1 -U flash:w:gen328.hex
```

After scratching my head, searching the Web, and finally doing some of my own experimentation, I discovered the problem. We (neither of us) didn't specify the baud rate. The SparkFun FTDI Basic board is a USB-to-serial translator. The Arduino Pro (ATMEGA328p) expects data at 57600 baud. Try this invocation and your setup will work:

```
Avrdude -c arduino -p M328p -P
/dev/cu.usbserial-A700e5J1 -b 57600 -U
flash:w:gen328.hex
```

The secret is the *–b 57600* which tells avrdude how to set the FTDI baud rate. Downloads pretty fast, too.

Q. I want to use simple radio control servos to run my robot. How fast do these servos go and can I use their own controllers to change their speed? I like using only one controller line to control the motor. Thanks.

— Tom, Council Bluffs, IA

A. Tom, I'm assuming that you want to know how fast your robot will go when using a hobby servo and not how fast the servo runs. Those two details are related, but not the same. There are a variety of hobby servos available and they have a variety of speeds and power. **Table 1** shows a small selection of popular hobby servos and a few that are more expensive to give you a quick range.

TABLE 1

Servo	Speed (4.8V)
Hitech HS225BB	0.14
Hitec HS311	0.19
Hitec HS422	0.21
Futaba S9002	0.18
Futaba S148	0.23
GWS S03N	0.23
Airtronics 94102Z	0.20

The speeds are given as time, in seconds, to move

the servo arm 60 degrees. You can then find the speed to move a full circle (after you have performed the "continuous rotation" surgery) or 360 degrees. I have given the speeds the servos will move at 4.8V since that is closer to the 5V that you might be powering the servos at. If you want a faster speed, use 6V and they go about 20% faster.

To find the time it takes a servo to spin a full circle (approximately), multiply the speed given above by 6 (6 x 60 = 360 degrees). This will give you *seconds per rotation*. To find *rotations per second,* take the inverse of that. In other words:

$$Rotations / Second = 1/(speed * 6)$$

This is only half of your question, however. How fast will your robot go? That depends on the diameter of your wheel. Assuming that your servo is strong enough to move your robot (which is ANOTHER topic), the speed of your robot will be the *Rotations per Second* times your wheel rollout. "What is wheel rollout?" you ask? Wheel rollout is the distance travelled by a single rotation of your wheel. This is the perimeter of the circle described by your wheel which is:

$$D = \pi d$$

where D is the distance travelled (in the units you chose) and d is the diameter of the wheel (in those same units).

Example: If we were to use a Lynxmotion "Sticky Servo Tire" of 2.75 inches diameter, then the wheel rollout will be roughly 8.64 inches. To put it all together then, we'll use this wheel with the Hitec HS311 hobby servo to get:

$$8.64 \times \frac{1}{(0.19 \times 6)} = 7.58 in / \sec$$

which is a comfortable table top speed. If you want to go faster, use a faster servo, bigger wheels (within reason), or higher voltage (also within reason).

Now for the last of your questions ... can you control the speed of the motor with the standard circuit board? Yes. The common analog hobby servo has about six speeds forward and reverse that can be attained with even crude eight-bit pulse generators on our hobby microcontrollers. All of those speeds are within about 5% of the center position which is the 1.5 ms pulse width. Experiment with yours and see what speeds you can find. Since most of us use about three speeds for our robots (off, slow, fast), this should be all you need for controlling the speed of your robot.

Q. I want to use a bend sensor to control the movement of a servo. I'm using an Arduino; how do I do that?

— RoboBob

A. RoboBob, reading a bend sensor on an Arduino has been done a few times; Google for some of them. It is basically just reading a resistor divider and the details of setting up the ADC registers.

I think that I'll approach this question on a broader scale. You want to control a servo using a bend sensor (always a fun project). A servo shows the jitter created by an imperfect world through that bend sensor pretty easily. Let's do a little more here and show you how to create a basic *sliding window* filter for your sensor that will really help reduce that jitter.

I have a project that I've been wanting to start for a while now that will do well to illustrate this concept. I'm going to hack an ancient Nintendo Power Glove to use as an R/C car controller for my son. I doubt that I'm inventing a better mousetrap, but it sure will look cool. The Power Glove has bend sensors in it already so I've got what I need to get started. I've measured these sensors, and they range between 70K and 400K ohms while bending. Reading the ATMEGA328 datasheet, we see that the ideal input impedance is 10K or less. But, since this is a slowly changing signal and I don't need great precision I'll punch ahead anyway and see how it works.

To handle this project, I'll just use the Arduino Pro 328 from our first question, since I already have it on the bench! I'll use a bend sensor from my Power Glove and a

Listing 1: Input a new filter value.

```
void FilterInput(int in)
/*
 * put in a new reading
 */
{
        int n;

        if (((in - bendFilter.res) > bendFilter.jmp) ||
                ((bendFilter.res - in) > bendFilter.jmp))    {
                bendFilter.n = 0;    // clear and start over
        }

        if (bendFilter.n == FSIZE) {                // Slide the window
                for (n=0; n<FSIZE-1; n++) {
                        bendFilter.window[n] = bendFilter.window[n+1];
                }
                bendFilter.window[FSIZE-1] = in;
        }
        else {
                bendFilter.window[bendFilter.n] = in; // Fill array
                bendFilter.n++;
        }
}
```

Listing 2: Doing the average.

```
int  FilterCalc(void)
/*
 * Calculate our results and return the current sliding
 * window average.
 */
{
        int n;
        int sum=0;

        for (n=0; n<bendFilter.n; n++) {
                sum += bendFilter.window[n];
        }
        bendFilter.res = sum/bendFilter.n;

        return(bendFilter.res);
}
```

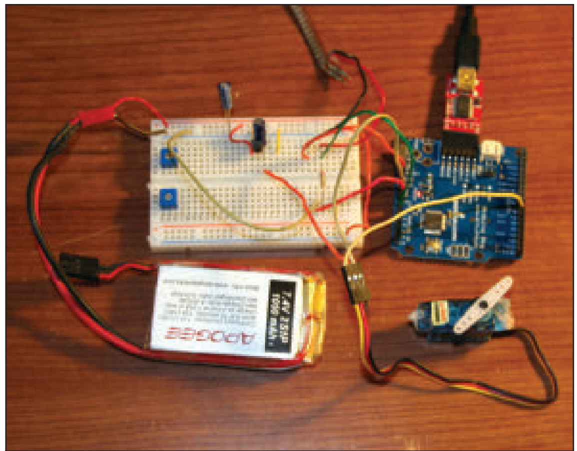


Figure 2. Bend sensor and servo hardware.

Listing 3: Using the filter.

```
int main(void)
{
        int n;

        initialize();                   // Initialize the board
        FilterInit(20);                 // This seemed to work well

        waitms(2000);
        printf("hello\r");

        while(1)
        {
                // Blink an LED on I/O 13 to show we're running
                PORTBbits.b5 = ~PORTBbits.b5;
                // Do something while wasting time
                for (n=0; n<10; n++) {
                        waitms(50);
                        // start sample/convert on ADC0
                        ADCSRA |=0x40;
                        // wait until ADSC clears
                        while (ADCSRA & 0x40);
                        // enter next value in the filter
                        FilterInput(ADCH);
                        // servo position 90 to 200 more-or-less
                        servo = FilterCalc();
                }
                // print our running average
                printf("read: %d\r",FilterCalc());
        }
}
```

little Dynam micro servo that pulls too much power. This latter is important since the proto board that I'm using has a 78L05 regulator only good to about 100 mA current. I'm not going to go into how to make a voltage divider or how to connect a servo to an Arduino; those topics can be *Googled* easily enough. The focus of this answer will be how to keep that servo from jittering in our use.

A sliding window filter is simply a way to use an average value of the last *n* raw values read instead of the raw value directly. This type of filter tends to smooth out minor noise in the signal that would otherwise have your reading bouncing around between numbers that are close together. When your window has "filled up," you bump the oldest reading off the list and enter the most recent one at the end.

Listing 1 shows the code for my sliding window filter. When we initialize the filter, we set the Jump Value (bendFilter.jmp) to the value we want to use to reset the filter. In other words, a difference from the average so great we just want to go there so we don't slow the response down when we want a large change in the value to be seen. The second part of this code either fills the window array or drops the oldest value off one end and puts the newest value in the last slot.

In **Listing 2**, we see how we do the average of our windowed values. As you can see, nothing scary here. Finally, in **Listing 3** is the source for the main loop showing how the filter is initialized and used in my little demonstration application.

Figure 3. The controlling hand.

In **Listing 3**, we initialize the filter with the function *FilterInit(20)*. This means that if any value is put into the filter that is 20 or more different from the current average, we will reset the filter and start a new average. Try various values. A "1" will essentially turn the filter off, a "100" will probably never jump and clear the filter. In my experiments, I found the value 20 to give me the best compromise between response and stability. You may find something that you like better; it depends on the application really.

No project is complete without a picture of the hardware, so **Figure 2** is my setup. I have a SparkFun Arduino Pro 5V board with the ATMEGA328p on it and a SparkFun FTDI Basic USB-to-serial board to program it. I used avr-gcc in an Eclipse IDE and avrdude 5.8 to program my board. Since — as most of you know by now — I am a Macintosh OS X user, all of my environment is on the Mac. I used good ol' Zterm as my terminal emulator so if you use something else, you'll want to experiment with what you need to get a CR/LF at the end of your printf lines.

You can see the bend sensor partially at the top of **Figure 2**; the proto board was used to just connect the sensor and the servo.

Oh, in case you're interested, **Figure 3** is what will eventually house the electronics to my Power Glove R/C controller. Like I said before, it probably isn't all that innovative, but for a nine year old it should be very cool. I'll include more photos and information as the project progresses.

Well, that's all for this month. I hope that I've stimulated your interest and motivated you to go out and try something new. The source code for the third question can be found with the article downloads at the link provided here. As always, if you have a question for Mr. Roboto, drop me a line at roboto@servomagazine.com and I'll be happy to try to answer it! Have fun and keep building robots! **SV**

Robotics Showcase

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

MARCH

6-10 **APEC 2011 Micromouse Contest**
Fort Worth Convention Center
Ft. Worth, TX
High-speed, tiny robots solve a maze as fast as possible. The APEC conference runs March 6 through 10. The Micromouse contest will be on March 7th at 8 pm.
www.apec-conf.org

11-12 **AMD Jerry Sanders Creative Design Contest**
University of Illinois at Urbana-Champaign, IL
Robots must identify and collect colored Nerf balls.
http://dc.cen.uiuc.edu

12-13 **RobotChallenge**
Vienna, Austria
Events include Micro Sumo, Mini Sumo, Standard Sumo, Parallel Slalom, and Slalom Enhanced.
www.robotchallenge.org

19 **Manitoba Robot Games**
Tec Voc High School, Winnipeg, Manitoba, Canada
Lots of events including Standard Sumo, Mini Sumo, Mini-Tractor Pull, Super Scramble, Line Following, and Robo-Critters.
www.scmb.mb.ca

22-24 **DTU RoboCup**
Technical University of Denmark
Copenhagen, Denmark
Robots compete on a varied course that includes line following and wall following.
www.robocup.dtu.dk

26 **Harrisburg University Pennbots**
Harrisburg University, Harrisburg, PA
Events include both remote-control combat and autonomous maze solving.
http://web.me.com/wjbechtel/Robot_Competition/Welcome.html

26-27 **Robocore**
Campos do Jordão, São Paulo, Brazil
Remote-control vehicle combat events.
www.robocore.net/modules.php?name=GR_Eventos&evento=10

APRIL

2 **CIRC Central Illinois Bot Brawl**
Lakeview Museum, Peoria, IL
Lots of events include 3 kg Sumo, 500 g Sumo, LEGO Sumo, line following, maze following for autonomous bots, and several combat events for remote control.
http://circ.mtco.com/competitions/2011

2-3 **Trenton Computer Festival Robotics Contest**
College of New Jersey, Ewing Township, NJ
Not sure what the specifics are yet, but based on last year's event, they'll have a variety of robot demonstrations and contests.
www.tcf-nj.org

9-10 **Trinity College Fire Fighting Home Robot Contest**
Trinity College, Hartford, CT
Autonomous robots must navigate through a mock house, then locate and extinguish a candle in the shortest time possible.
www.trincoll.edu/events/robot

10 **Robotics Innovations Competition and Conference**
Woburn, MA
University students must engineer robots to solve real world problems. This year, they must demonstrate robots that use unconventional means of mobility to navigate through

environments that are inaccessible to wheeled and tracked robots.
http://ricc.wpi.edu

14-16 **National Robotics Challenge**
Marion, OH
A variety of student robot events including Robo Hockey and Sumo.
www.nationalroboticschallenge.org

14-16 **VEX Robotics World Championship**
Kissimmee, FL
High school and university VEX teams compete with robots that are operated in both autonomous and remote-control modes.
www.vexrobotics.com/competition

15-17 **RoboGames**
San Mateo Fairgrounds, San Mateo, CA
This event includes FIRA, BEAM, Mindstorms, and lots of other events for autonomous and remote-control robots.
www.robogames.net

23 **Baltic Robot Sumo**
Klaipeda, Lithuania
Autonomous robots compete in Mini Sumo for a traveling cup prize.
www.balticrobotsumo.org

30 **The Tech Museum of Innovation's Annual Tech Challenge**
Parkside Hall, San Jose, CA
This year's challenge is called "Explore the Volcano."
http://techchallenge.thetech.org

MAY

7 **RoboFest**
Lawrence Technological University, Southfield, MI
Game Competition: two autonomous robots work together. Robot Exhibition, RoboFashion Show, Mini Urban Challenge, Fire Fighting, and VEX.
http://robofest.net

9-11 **FIRA Robot World Cup**
Dubai, UAE
HuroSot — Humanoid Robot World Cup Soccer Tournament; S-HuroSot — Single Humanoid Robot World Cup Soccer Tournament; RoboSot — Robot World Cup Soccer Tournament; S-RoboSot — Single Robot World Cup Soccer Tournament; MiroSot — Micro Robot World Cup Soccer Tournament; S-MiroSot — Single Micro Robot World Cup Soccer Tournament; NaroSot — Nano Robot World Cup Soccer Tournament; S-NaroSot — Single Nano Robot World Cup Soccer Tournament; KheperaSot — Khepera World Cup Soccer Tournament; and S-KheperaSot — Single Khepera World Cup Soccer Tournament.
www.fira.net

9-13 **ICRA Robot Challenge**
Shanghai, China
This event includes a Planetary Exploration Challenge, Human-Robot Interaction Challenge, and Virtual Manufacturing Automation Challenge.
http://icra.wustl.edu

23-26 **NASA RASCAL Robo-Ops**
Johnson Space Center, Houston, TX
Student competition in which teams compete to build planetary rovers that will have to navigate the JSC "Rock Yard" by teleoperation.
www.nianet.org/RASCAL/RoboOps/index.aspx

JUNE

2-4 **ION Autonomous Lawnmower Competition**
Beavercreek, OH
Autonomous lawn mowing.
www.automow.com

3-6 **AUVS International Ground Robotics Competition**
Oakland University, Rochester, MI
Autonomous ground vehicle navigates outdoor obstacle course within a prescribed time while staying within a 5 mph speed limit. Autonomous (emergency RC control and manual stop required). Open to college students only.
www.igvc.org

25-26 **International Autonomous Robot Contest**
San Diego County Fairgrounds, San Diego, CA
Autonomous navigation around fixed obstacles.
www.iaroc.org

NEW PRODUCTS

GEARBOXES

CIM-U-LATOR Gearboxes

Banebots' new CIM-U-LATOR gearboxes allow users to take RS-550, RS-540, and RS-775 motors and mount them anywhere a CIM can be mounted. Highlights include:

- Simple, small, lightweight: 1.5" x 2.5" x 1"; less than one pound.
- Two initial configurations: one RS-775, one or two RS-550/540.
- Fixed ratio of 2.7:1 giving output speed close to a CIM.
- Output shaft is supported by two ball bearings making them ideal for stand-alone use, as well as CIM-U-LATION.
- Introductory price of under $30.

The CIM-U-LATOR gearbox mounts one or two RS-540/RS-550 sized motors. The output shaft and face mounting are the same as a CIM motor. The gearboxes are made from two aluminum blocks with a cavity machined between them for the gears. Each block has a ball bearing to support the output shaft; output gear is located on the shaft between the ball bearings. The two blocks are assembled with four 6-32 socket head cap screws — one in each corner. Each block has two #10-32 mounting holes on one of the 2.5" sides. When the blocks are assembled, the side mounting holes form a 0.75" square centered on the bottom of the gearbox. The blocks are symmetrical so that one block can be rotated 180 degrees giving mounting holes on top and bottom. The two #10-32 face mounting holes are 1.25" from the top/bottom of the gearbox (centered) and 1" from the center (two inches apart; same as CIM).

Physical Specifications:

- Type: Spur
- Reduction: 2.7:1
- Stages: 1 (10 to 27 tooth)
- Gear Material: Steel
- Gear Pitch: 24
- Weight: 16 oz
- Length: 2.5 in
- Width: 1.5 in
- Depth: 1 in
- Shaft Diameter: 8 mm
- Shaft Length: 1.56 in
- Shaft Key: 2 mm
- Mounting Holes (6): #10-32

Some assembly is required. Gearboxes are delivered assembled and tested; however, they will need to be disassembled to grease and mount the motor(s) prior to use. No special tools are required; typically two hex wrenches. A small arbor press or bench vise is useful for pushing the pinion onto the motor shaft. The gearbox must be greased prior to use. Grease is not included and must be purchased separately.

The gearboxes include one mounting kit which contains everything required to mount one motor including pinion, screws, and lock washers to attach the motor to the gearbox (motor not included). The correct pinion for this gearbox is the S24P-GMC1-32; additional pinions can be purchased separately or as a complete mounting kit.

For further information, please contact:

BaneBots Website: **www.banebots.com**

MOTOR CONTROL/ CONTROLLERS

Motor Controller

The USMC-01 stepper motor controller IC from Images Scientific is a chip that can operate as a slave or master. As a slave, it runs under another microcontroller (or PC) or it can operate in auto-run (master) mode which allows the user to add a few switches and run the stepper motor manually. RPMs of stepper motors are switch selectable.

This stepper motor chip generates control signals that can be used with both unipolar and bipolar stepper motors with appropriate drivers like the L298 and L293.

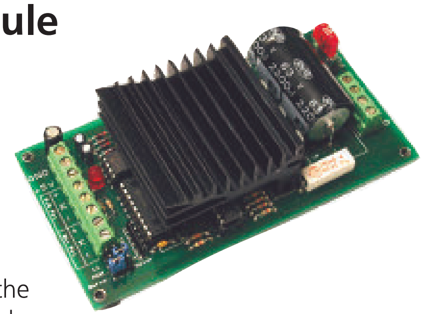
Features:

- General/hobby purpose stepper motor controller.
- Compatible with four phase unipolar/two phase bipolar motors.
- Master/slave mode — stand-alone free running mode.
- Eight RPM selections in free running mode.
- Compatible w/drivers - L298, L293, discrete transistors.
- Half/full wave: step modes, direction control, enable.
- Power on reset initializes controller.

For further information, please contact:

Images Scientific Instruments Website: **www.imagesco.com/stepper/usmc.html**

Programmable Motor Control Module



This unique board available from Blue Point Engineering will allow the intermittent directional control (CW/CCW) of a DC motor. The time that the motor will travel in each direction is programmed into the board and is stored in non-volatile memory. There are limit switches that — when activated — will stop the motor.

The board can be triggered by either a dry contact closure at the remote trigger terminal, or by pressing the adjacent tactile switch. Once triggered, the motor will travel in the first direction for the programmed time, wait for a programmed time, travel in the opposite direction for a programmed time, wait for another programmed time, and then the board will wait for the next trigger event. Alternatively, if Auto Loop is programmed to be on, the board will then wait for a programmed amount of time and then automatically repeat the entire process. All programmed times can range from one second to 60 minutes. Programming is accomplished via a four line x 16 character LCD display and four programming buttons.

Power to the board is via a 2.1 mm barrel-type connector or via the adjacent terminal block. Power to the motor is applied via another 2.1 mm barrel-type connector or via the adjacent terminal block. LEDs on the board will indicate that power is present. There is a small slide switch at the top of the board that will allow you to turn the LCD backlight on and off. LEDs next to each limit switch terminal block will indicate that a limit switch is in the open condition. There is also an SS relay contact that will close for two seconds at the beginning of motor travel in each direction. A LED next to the SSR contact terminal block will indicate that the relay is active and the contacts are closed (motors, power supplies, and limit switches are not included). Board size is 6-1/4" L x 3-7/8" W x 1-1/4" H. The motor relay is rated at 10 amps DC. Duo power supplies are required. Board electronics are 9-12 VDC. Motor power is 5-24 VDC.

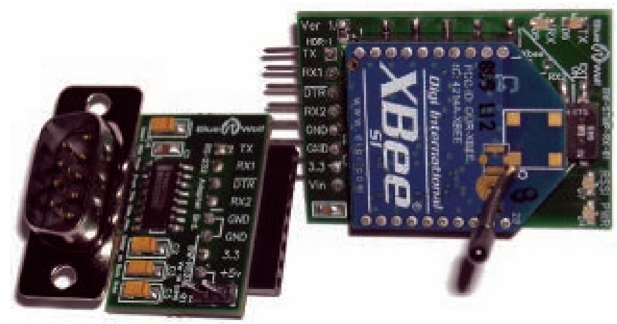
For further information, please contact:

Blue Point Engineering	Website: **www.bpesolutions.com**

SYSTEMS AND PLATFORMS

BASIC Stamp Wireless Download

Blue Wolf, an engineering and design company, is introducing a new line of wireless robotic products. Their newest product is FlashFly which is an innovative



system that allows a user to remotely download BASIC Stamp programs to Parallax's Stamp modules or interpreter chips. FlashFly consists of three interdependent modules: the remote receiver board; an optional RS-232 adapter board; and the XBee USB base transmitter board.

FlashFly's wireless capabilities — in coordination with common Series 1 XBee modules — will allow a user with a BASIC Stamp mobile robot or stationary platform to modify his or her program remotely. FlashFly eliminates the tedious task of having to connect a robot to a computer before any programming changes can be made. FlashFly also enriches a user's learning experience by providing instant feedback data to the DEBUG terminal screen which allows a user to evaluate his or her program flow and I/O data remotely. FlashFly can also be used in conjunction with any other microprocessor platform that needs to transmit data wirelessly.

Blue Wolf designed FlashFly with versatility and convenience in mind for both hobbyists and educational users. The (1x8) 0.1" inline header makes breadboard use simplistic. Alternatively, an RS-232 converter board can be plugged in for direct connection to a DB9 female connector on a remote or mobile robot platform. Furthermore, FlashFly can be used for more advanced XBee experimentation by directly connecting additional wires or headers to the two rows of 0.1" plated holes on FlashFly. Any existing Series 1 XBee modules can be easily reconfigured to work with the flexibility and modularity of FlashFly. FlashFly features include:

- Onboard 3.3V regulator.
- Small footprint for mobile robotic applications or fixed platforms.
- LEDs for indication of TX, RX, RSSI, and POWER.
- Uses Parallax's BASIC Stamp Editor software for program downloading (no additional steps required).
- Two rows of 11 plated through holes with 0.1" spacing allow the option of soldering jumper wires or headers (not included) for easy access to the remaining XBee module pins for more advanced designs.
- Designed for the option of using XBee Pro modules if a larger wireless range is required.
- Low cost, flexible use system that can be used on

Continued on page 51

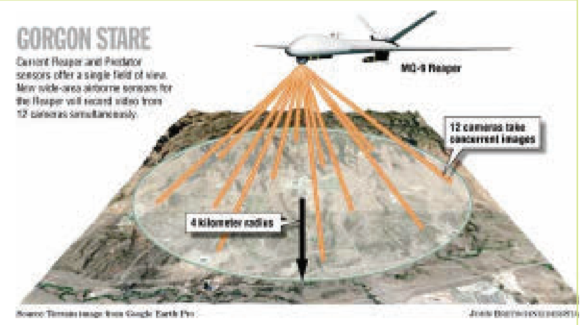
bots IN BRIEF

NEW SCOOBA REALLY CLEANS UP

iRobot has a new Roomba designed for that "special" area around your toilet or other tight places. The 3.5 x 6.5" Scooba 230 will be available at a price of $299.99, and it has a three-stage cleaning system that washes, scrubs, and squeegees floors.

The robot's cleaning system neutralizes up to 97 percent of common household bacteria and it holds enough cleaning solution to scrub up to 150 square feet of linoleum, tile, or sealed hardwood floors in a single cleaning session.

STARE INTO MY ...



The $17 million Gorgon Stare airborne camera array has made it possible for soldiers on the ground to see live images when searching for enemy movement. It consists of nine cameras that can send up to 65 different shots to separate users and it has the capability of seeing an entire city of up to 4 km. However, the military admits that it is only as good as the human intelligence using it.

Gorgon Stare was conceived, designed, and developed in less than three years by prime contractor Sierra Nevada Corp. and USAF's 645th Aeronautical Systems Group — a rapid acquisition arm also known as Big Safari. It offers exponential expansion in the scope, amount, quality, and distribution of video provided to ground troops, manned aircraft crews, ISR processing centers, and others,

Gorgon Stare's payload is contained in two pods slightly larger than, but about the same total weight as the two 500 lb GBU-12 laser-guided bombs the Reaper carries. The pods attach to the inside weapon racks under the wing. One pod carries a sensor ball produced by subcontractor ITT Defense that protrudes from the pod's bottom. The ball contains five electro-optical (EO) cameras for daytime and four infrared (IR) cameras for nighttime ISR, positioned at different angles for maximum ground coverage.

GET CARTED AROUND

Those little ZMP robot cars that came out in 2009 have spawned a giant new family member — the "RoBoCar."

It's really more like a golfcart, but it's something that you could realistically drive around in. Or, be driven around in since it's capable of autonomous speed adjustments, braking, and steering. The deluxe model comes complete with a stereo camera, an IMU, and laser rangefinders, and it only costs about $80k. If you'd rather take care of everything but speed control yourself, a stripped-down version with fewer autonomous capabilities will only set you back about $30k.

bots IN BRIEF

TALK TO CHUCK

After suffering his own bout of road rage with his GPS unit, Cambridge University Professor Peter Robinson decided to create Charles. The head and shoulders device rides shotgun, reads facial cues and tone of voice of the driver, and sympathetically gives directions. Robinson claims that the robot is about 70% accurate at reading emotions and wants to further his development to be able to lower the radio volume or avoid instructional repetition.

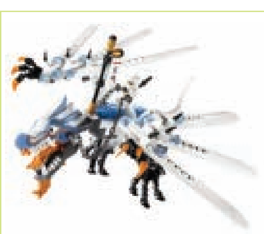
ALL TERRAIN, ALL THE TIME

University of Pennsylvania's KodLab has developed a new version of their RHex wheeled/legged robot called X-RHex. X-RHex is about the same size and weight as RHex, but it's stronger, more durable, and has a longer run-time of up to two hours. It's also got a couple mil-spec rails mounted on top, along with standardized electrical connections. X-RHex doesn't seem to manage the same speeds exhibited by other leg/wheel hybrid robots such as Whegs, but its strength is in its adaptability and the way it can make it through basically any sort of terrain — even things that would challenge conventional wheeled or tracked robots.

X-RHex is designed with a modular payload architecture to support a variety of research needs. Six powerful motors actuate compliant legs, allowing X-RHex to traverse a wide variety of terrains, including asphalt, grass, sand, mud, and rocks. By using a mil-spec rail mounted interface and standard electrical connections, X-RHex can support a wide variety of payloads, making it a mobile "laboratory on legs."

ELECTRIC PERSONALITY

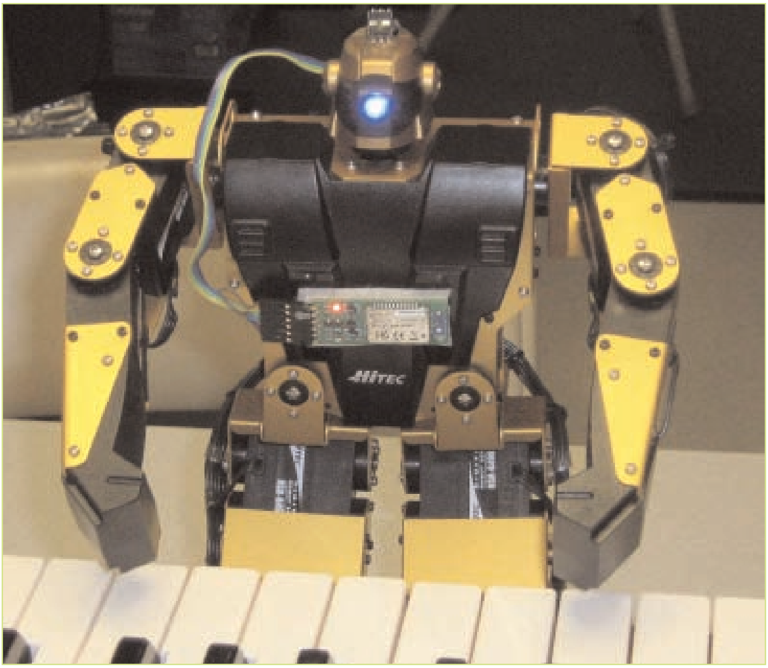
Kikkerland's Electro Man is a four plug multi-outlet with arms and legs that have three-pronged sockets. The plugs are grounded and a circuit breaker is included. At a size of 14 x 12 x 1/2" and with a 28" cable, Electro Man is controlled by an on/off switch and smirks as he powers up your electronics and appliances.



ATTACK OF THE ICE DRAGON

LEGO has entered the realm of Ninjago with its new Ice Dragon Attack Set. With a Krazi skeleton figure, the 11" long Dragon can spit iceballs, move its legs and tail, and has bendable/folding wings. A total of 158 pieces includes two minifigures, ice shurikens, and two weapons.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.



MUSIC MAN

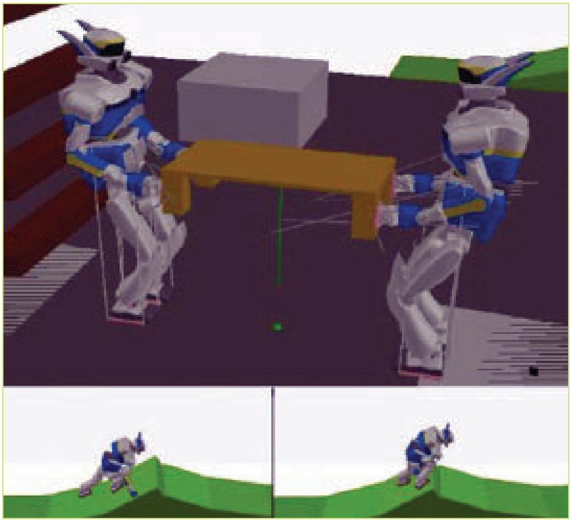
Drexel University's Alyssa Batula and Dr. Youngmoo Kim taught a Robonova several tricks, including dancing, tapping a tambourine, and playing the piano with only two digits. They used this particular humanoid because it can mimic human gestures and is both rugged and inexpensive. This particular robot was useful to them for several reasons:

- The RoboNova kit can be configured as a humanoid, so it can mimic human gestures.
- This robot is both rugged and inexpensive. It is unlikely that it would be damaged beyond repair in testing, and in the worst case, they would easily buy another one. This allowed the team to prototype new and potentially destabilizing algorithms without worrying about destroying the robot.
- Each limb allows a fine degree of control, enabling precise fine-tuning of gestures.
- The programming environment is straightforward.
- The robot provides rudimentary sensors for feedback, such as a gyroscope.

The Robonova also has two major disadvantages. First, it is a relatively simple robot and cannot mimic human motions all that precisely. Second is that the computer onboard the robot is underpowered. This means that most of the processing must take place on an offboard computer, so the algorithms can run in real time. Nevertheless, they enabled the RoboNova to perform a variety of tasks, such as dancing to live music and playing the piano.

TEAM EFFORT

Researchers including Sebastien Lengagne, Karim Bouyarmane, and Abderrahmane Kheddar at AIST are trying out a novel approach to navigating cluttered environments. Rather than seeing household objects as obstacles to be avoided, they are programming a robot to use them for support. For example, if a human were to look under a desk they'd probably hold onto it for support, but a robot is usually programmed to do these sorts of movements without touching anything. In physical tests, the researchers are keeping things relatively simple; the HRP-2 Promet can steady itself on a table when sitting down in a chair or when taking a long step. In simulation, they're able to get a bit more ambitious, like having two robots cooperatively manipulate an object or climb a steep incline.

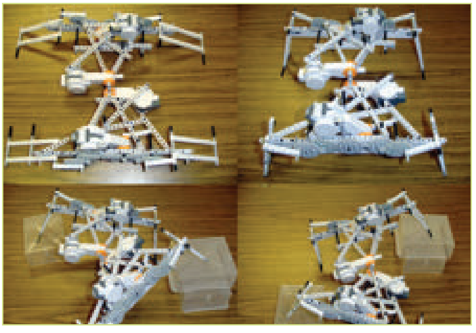


Collisions with objects (including other robots) and attempting to balance on them together pose a number of problems. For example, a robot needs to know what objects can physically support them. A robot's joints could be damaged or the robot may slip if they aren't properly supported. The researchers are developing simulation software that determines contact points between the robot and any other object and the environment, while also factoring in the effects of gravity and actuator torques, ensuring they are stable. The resulting work could also be used to help robotic manipulators grasp objects, or even work together to manipulate objects too large for one robot to carry alone.

CANDY MAN

A while back, TMSUK teamed up with other robot-related businesses in Kyushu including Yaskawa, Simex, and Kyushu University to develop the T-82. Surprisingly, the robot's main claim to fame was not its medieval appearance (which included a sword and shield), but rather the ultra-compact, lightweight actuator and control unit used in its multi-fingered hands. The robot was originally presented scooping candy into a plastic bag for an audience of schoolchildren, and also held a bottle in one hand and twisted the cap (to open and close it) with the other.

The robot's hands had three fingers — each with three joints — measuring about 20 cm from knuckle to fingertip. The miniature actuator weighed approximately 1 kg and had a fingertip force of 1 kg. At the time, Yaskawa claimed they had developed the smallest robotic hands in its class which could help robots interact with their environment better in the future. The T-82 appears to have a single pan/tilt camera for a head, arms with six degrees of freedom, and a mobile base.



GIANT LEAPS

Josh Bongard of the University of Vermont has determined that if you build a simple robot, it will eventually evolve by itself. Using LEGOs, he built actual bots and simulated ones that changed over time. Eventually, the simple ones could walk better than those with fixed bodies. Bongard's research is supported by the National Science Foundation and their study on evolutionary robotics.

Bongard's bots are like tadpoles that become frogs, changing their body forms while learning how to walk. Over generations, his simulated robots have evolved, spending less time in "infant" tadpole-like forms and more time in "adult" four-legged forms.

As mentioned, these evolving populations of robots were able to learn to walk more rapidly than ones with fixed body forms. In their final form, the changing robots had even developed a more robust gait; they are better able to deal with being knocked with a stick, for example, than the ones that had learned to walk using upright legs from the beginning.

DIVER DOWN

Photo courtesy of Walker Smith.

Researcher Walker Smith of the Virginia Institute of Marine Science, College of William and Mary, has been conducting shipboard studies of biological productivity in Antarctica's Ross Sea for the last three decades. Now, he's letting underwater robots do some of the work.

Smith and graduate student Xiao Liu are using a two-year grant from the National Science Foundation to deploy and test a free-swimming underwater glider in the frigid waters of the Ross Sea near the US McMurdo Research Station. The grant also funds efforts by fellow VIMS professor Marjorie Friedrichs to use glider data to help improve computer models of the Ross Sea's physics and biology.

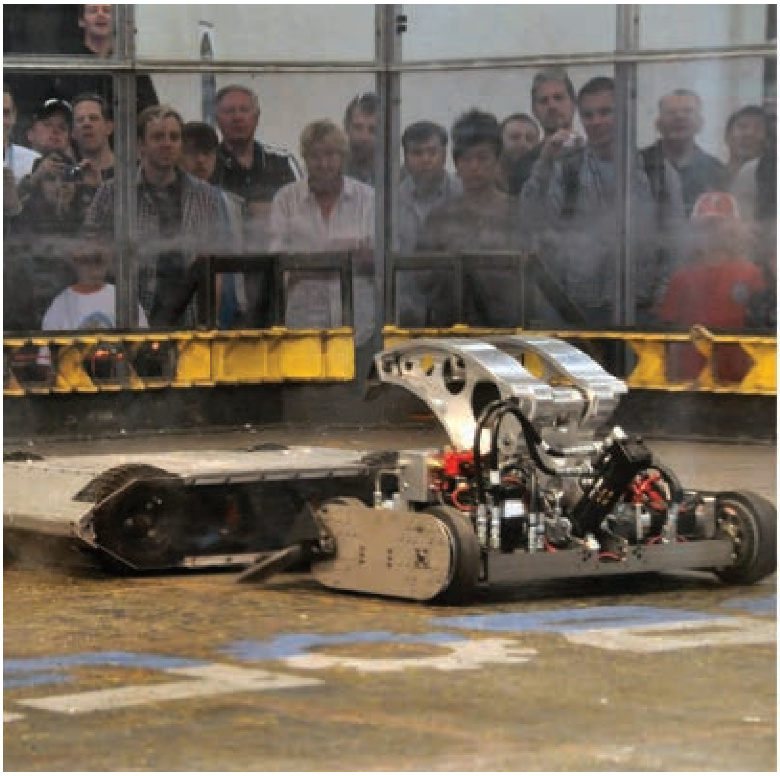
Smith deployed the team's glider, SG503 — also know as the Ice Dragon — for its first mission on November 29, 2010. He and colleagues, including investigators from Old Dominion University, launched the 114 pound vehicle through a whale breathing-hole, and then directed it into the open waters of the "polynya" that forms each austral summer when seasonal sea-ice melts from the Ross Sea.

The launch — at a latitude of 77°S — is the most southerly glider deployment ever. A short (and unintentional) jog off course also made it the first-ever glider to successfully dive beneath the Ross Ice Shelf. As of January 19, 2011 the glider had completed 783 dives to depths as great as 700 meters (2,330 feet), traveling a total of 1,402 kilometers (871 miles). It is scheduled to continue yo-yoing back and forth across the Ross Sea polynya until the researchers retrieve it in early February.

Each of the glider's dives lasts about 120 minutes, during which sensors on its fiberglass hull measure water temperature, salinity, levels of dissolved oxygen, and chlorophyll concentrations (the latter is a measure of photosynthesis and phytoplankton abundance). At the end of each dive, the glider flips its tail into the air so that its antenna can send the collected data to researchers and it can receive guidance for its next dive. Data is transmitted via the Iridium satellite network.

ROBOGAMES

“World’s Largest Robot Competition” - *Guinness Book of Records*
“Top 10 Video Highlights” - ESPN *SportsCenter’s Play of the Day*
“The best robots compete in RoboGames, just as the best athletes train for the Olympics” - *Discover*
“The Best Ten North American Geek Fests” - *Wired*



RoboGames is the olympics for robots – a three-day event in the San Francisco Bay Area that brings the smartest humans and best robots from around the world to compete in a wide array of robotics oriented events (over 40 countries have participated in past events.)

Cart-wheeling androids, combat robots, autonomous vehicles, LEGO robots, soccer bots and even cocktail-mixing barbots – there’s something for everyone at RoboGames! The event also features demonstrations by leading robotics industry designers and engineers, kinetic art exhibits, and the latest in tech products, gear, and innovation.

RoboGames began as an enthusiastic experiment in robotic cross-pollination, when dozens of disparate, well-established robot competitors were placed under the same roof. Bringing together builders from acclaimed fields such as combat robotics, robot soccer, sumo, fire-fighting, androids, and kinetic art, RoboGames enables robot builders to exchange ideas and share their knowledge and experience with each other. Varying disciplines now learn from one-another and the event has grown into a fantastic multi-layered, multi-cultural experience like no other. The best part is that RoboGames is completely open–anyone can compete: competitors have been garage builders, K-12 school teams, professional engineers, and university researchers. Come see the future evolve!

Educational Outreach: In addition to the 60 adult events, RoboGames sponsors 10 different “junior league” events for kids in K-12, which are free for kids to compete. University students also have the opportunity to publish and present research papers.

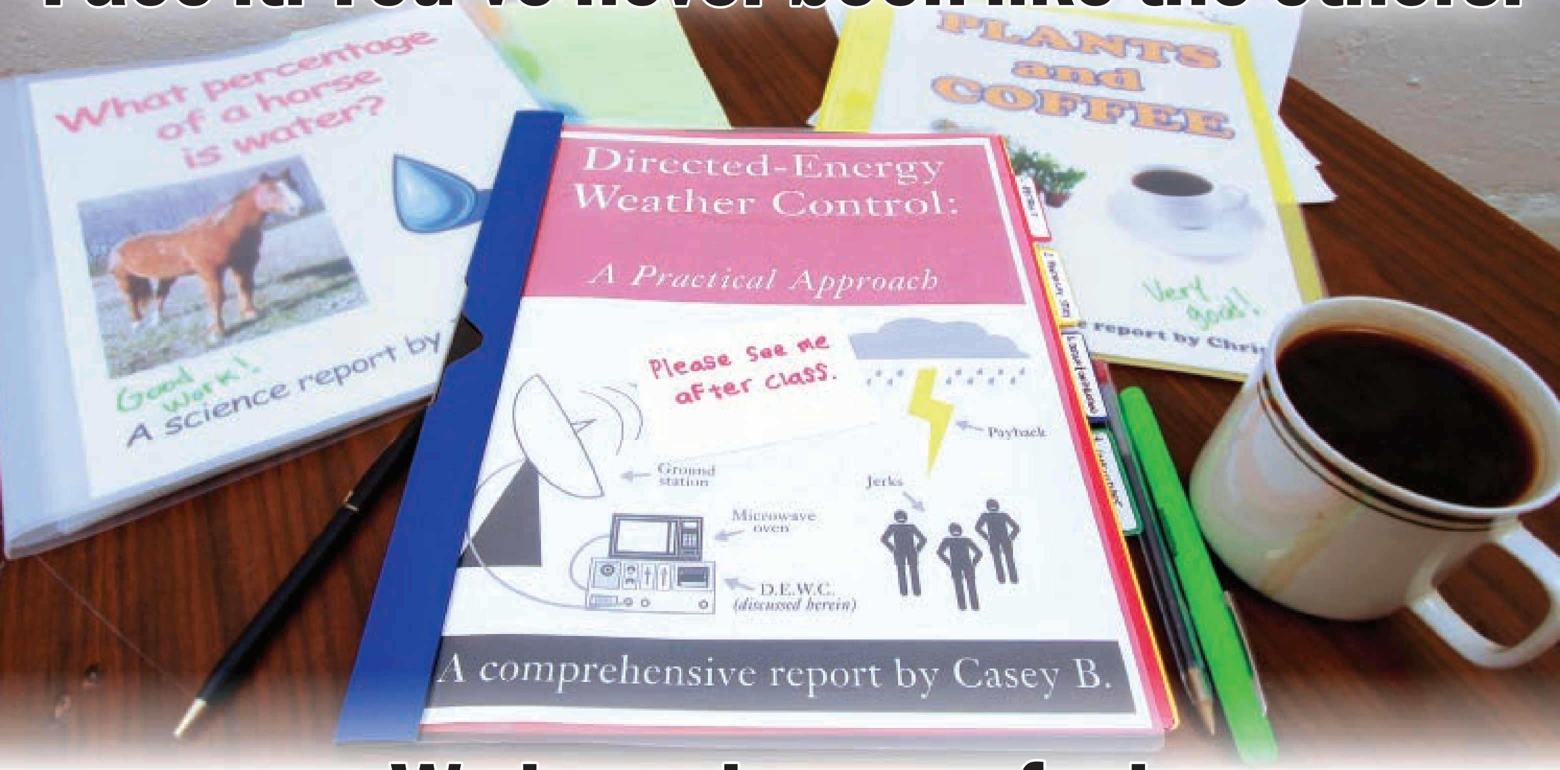
Sponsors: Your company can reach millions of people around the world by sponsoring RoboGames. Tens of thousands of people attend the event in person, and millions are reached from the media-exposure – print, web, radio, and television. Popular with techies, sports-fans and hipsters alike, RoboGames has something to offer every demographic.

Go on-line to find out more! Watch videos, get building tips, register to compete, buy tickets or sponsor the next event.

This years’ event will be taped by The Discovery Network for a TV show to be released this year! If you can’t compete or attend, be sure to watch for us on TV!

April 15-17th, 2011 - San Mateo, California - http://RoboGames.net

Face it. You've never been like the others.



We know how you feel.

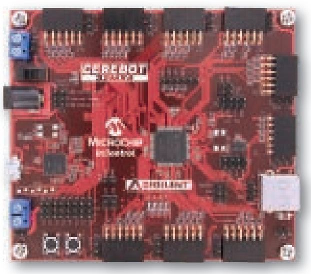
CEREBOT™ 32MX4

$74

*Enter the value code "**servo2011a**" on our website for this special introductory price*

A powerful PIC32® based microcontroller board, ideal for robotic experimentation in C, C++, and Assembler.

- **Microchip® PIC32MX460F512L**
- **80 MHz 32-bit MIPS processor**
- **512K Flash, 32K RAM**
- Eight R/C servo connectors, two SPI ports, two I^{2}C ports, two UARTs
- Nine 12-pin Pmod™ Connectors
- Five 16-bit timers with 5 input capture and 5 PWM outputs
- 16 channel, 10-bit, 500ksps A/D converter
- On-board USB Program / Debug Interface
- USB OTG Host / Device Capable
- USB Powered

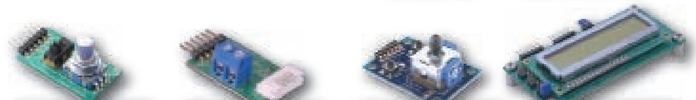


Pmod™ Peripheral Modules

Expand your designs

with sensors, actuators, and other I/O.

$9.99 - $29.99

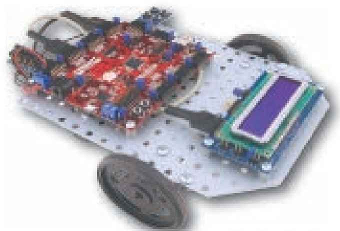


PmodENC	**PmodHB5**	**PmodJSTK**	**PmodCLS**
Rotary encoder module w/ integral pushbutton	2A H-bridge w/ feedback inputs	2-axis joystick w/ 3 pushbuttons	2-line character LCD w/ serial interface

RDK

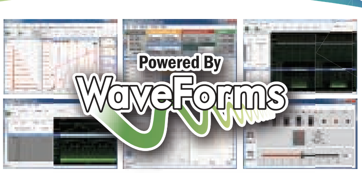
ROBOTIC DEVELOPMENT KITS

Complete kits include the Cerebot™ 32MX4, chassis, gearmotors, wheels, sensors, and software.



Starting at $149

ELECTRONICS EXPLORER™



Analog design revealed

A complete, circuit development kit that includes a breadboard, programmable power supplies, and powerful test instruments.

- 4-channel 40 MSa oscilloscope
- 2-channel arbitrary waveform generator
- 32-channel logic analyzer
- 32-channel digital pattern generator
  ...and more.

$399 (Academic)

Complete System!
Includes EE Board + Software

DIGILENT®

www.digilentinc.com

COMBAT ZONE

Featured This Month:

BUILD REPORT:

Building Combat Arenas: A Guide to Staying Safe

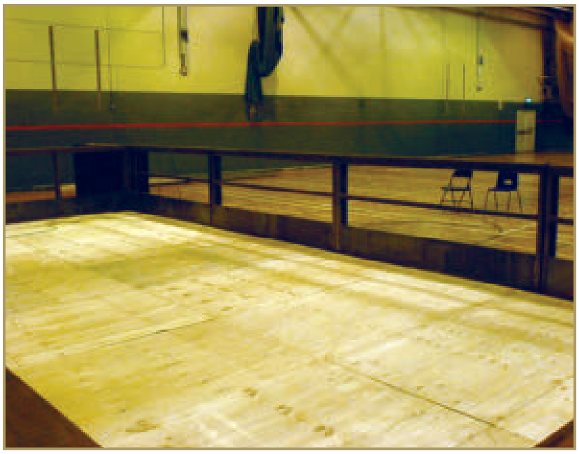
● by James Baker

Combat robots are dangerous; we can all agree about that. Anything designed to smash titanium will surely make short work of us mere humans. Event organizers face a battle of their own to contain our home-built terminators, and protect the fans who love to watch the carnage (but would not really like to be a part of it). As the robots get bigger, so do the challenges they pose. For anyone who has ever thought about building their own arena, here is a general guide. Just don't expect to find technical plans on this page, as designing your arena is something you should either do yourself, or have a suitably qualified expert do for you.

Seeing Right Through It

A universally accepted solution to the common problem of flying debris is the transparent screen. Usually made of polycarbonate, the screen must be an effective barrier between the public and the projectiles that a good fight will generate. Unfortunately, big plastic screens

An 8,000 lb all welded arena.



cost a good deal of money, and for the average builder wanting to create their own combat arena for testing, playing, private fun, and small public shows, it is often more cost-effective to have an open arena, and either limit the robots who use it or have everyone take cover.

Carefully cutting the welds to disassemble the arena.

Be Open About It

An open arena is not an unsafe option. For smaller robots without high power weapons, there should be no reason why an arena could not be built in a couple of hours from cheap interlocking chipboard panels and some 2x4s. I have built many of these over the years for youth clubs, scout groups, and military cadets, using only the most basic of tools, a bit of glue, and plenty of screws. This type of arena is a good solution to the main problem caused by small combat robots, as they scratch, scrape, and lay rubber on the arena floor. It is also a great way to learn how large washers can spread the loads, and where the forces are greatest on the walls and structure.

As we go up in robot size and/or power, the arena walls need to be a bit more substantial than soft wood, but a simple welded framework should be more than capable of taking the impacts if done properly. By tailoring the height of the walls to suit your preferences, you can ensure the robots do not get thrown out of the arena. Or, alternatively, you could set them just high enough to make it difficult but not impossible to throw your opponent out. We actually have a section of wall on our arena that opens up and becomes a hazard feature during the battle.

Remember also, that if you want to allow the excitement of "over the wall" outcomes, you must protect the surface they will fall on to with a "robot drip tray" of some kind — such as the wooden arena described previously — which has the added benefits of protecting the floor from the metal arena frame, and stopping the expelled robots from driving into the crowd.

The welded framework arena can easily be scaled up to cater for the largest of robots, but the costs involved will grow at an exponential rate because as the robots get bigger, the materials involved must be stronger, and far more must be used. The logistics of using a larger arena is also non-linear, with weight, handling, set-up time, storage, and maintenance being major considerations.

By bolting several smaller sections of the arena together, you can quickly assemble and disassemble the structure while making it manageable and easily storable. However, as the arena takes damage, flexes, and warps, these bolts may not fit so well or may snap, which can slow the process considerably. Bolts are also heavy and costly when used in the high numbers required on large multi-segment arenas, not to mention how easy it is to lose a few at the worst possible time. I once built an 8,000 lb combat robot arena for a client that did not have a single bolt in the whole structure. It actually was a dream to assemble, but it did require competent welding to join the 350 lb sections together inside the venue, and it involved carefully cutting and grinding the welds to take it back down after the event. The job of loading and unloading such weights can be a health and safety nightmare which is not something the general public should do without training and experience.

Arenas can be modular which allows for different sizes, shapes, and configurations, and is perfect for travelling with and putting on a show since you can configure your presentation to smaller venues. It is best to bolt this type of arena together, as it is a quieter, venue-friendly solution compared to welding. A fixed, welded structure suits larger, longer-term arenas at bigger events, or a semi-permanent garden or yard feature for the hard-core robot builder.

A lightweight, modular arena for smaller robots.

A very simple arena can be built by youngsters in two hours.

You would not be popular doing this to a gymnasium floor.

Pick Your Bot Preference

Up to this point, I have described the size, strength, and construction of the arena, but another major consideration to look at if you want screen-free fun is the robots you want to use. Spinners are not the best choice for unscreened combat, as they can throw anything across the arena, from a whole robot to a tiny bolt head. In the case of the latter, a bullet-sized object at a couple of hundred miles per hour is not going to make anyone's day if it hits them, and neither is a detached axe-head that may become a real pain in the neck ... perhaps literally.

Flippers may seem safe, but surely anything capable of throwing a 200 lb robot into the air is equally capable of sending a good chunk of metal into the audience, right? You can't even really feel safe if neither robot has a weapon because a couple of 200 lb machines grazing each other at full speed can easily flick a bolt head towards the viewers at unexpectedly high speed.

As I mentioned previously, you can get away with this if the robots are small enough and the kinetic energy available to any flying parts is low. Remember, as robots get bigger, the potential for high-energy debris grows quickly — especially if they are poorly designed and built. Simply countersinking bolts removes a lot of problems.

Ways to Serve and Protect

You can have a bunker for everyone to hide in or screens for people to hide behind, but you can never be sure that animals or pets are not going to wander in, or a well-wishing mother may bring everyone drinks mid-battle. There's even the most overlooked of possibilities: a piece of shrapnel could hit that one-in-a-million thing you really don't want it to hit, like a fire alarm button, exposed section of pipe, or electrical wire. Or, in a worst case scenario, your eyeball. I have seen a sheared bolt hit a bird in flight, so never assume something is too small or too unlikely to get hit.

There is another way to reduce the risks and this one may be the definitive guide to running an unscreened arena. Turn down your robot's power; reduce its speed and its pressure. Your opponent should do the same, and you both should ensure that the robots do not hit each other in areas likely to cause debris. By choreographing the fight — like entertainment-based wrestling — you can give a taste of the full combat event without the need for parts flying. This "fake" robot combat is the best way to make unscreened arenas work for the general public in my personal experience, but real fights with small or low power robots can also be possible with precautions.

Be sensible, stay safe, and if you are in any doubt, talk to the many professional event organizers out there. Use their knowledge and experience to make decisions about your arena and the robots that fight in it.

One final point, if you do decide to build an arena — large or small, with or without screens, real or fake fights, for the public or just for you — make sure you get insurance because the very act of building this "safety" structure may make you liable for any damage or injury while using it, so protect yourself in this respect. Visit **www.xbotz.com** for more information. SV

RioBotz Combot Tutorial Summarized: Brushless DC Motors

● Original Text by Professor Marco Antonio Meggiolaro; Summarized by Kevin M. Berry

Editor's Note: Professor Marco Antonio Meggiolaro, of the Pontifical Catholic University of Rio de Janeiro, Brazil, has translated his popular book — the RioBotz Combot Tutorial — into English. In January's issue, we summarized part of Chapter 5, dealing with brushed DC motors. Continuing the topic, this month we present in its entirety Marco's write-up on brushless DC motors. Marco's book is available free for download at: ***www.riobotz.com.br/en/tutorial.html****, and for hard copy purchase (at no profit to Marco) on Amazon, published by CreateSpace. All information here is provided courtesy of Professor Meggiolaro and RioBotz.*

Brushless DC Motors

A brushless DC motor is a synchronous electric motor powered by DC current, with an electronically controlled commutation system instead of a mechanical one based on brushes. Similarly to brushed DC motors, current and torque are linearly related, as well as voltage and speed.

In a brushless DC motor, the permanent magnets rotate while the armature windings remain static. With a static armature, there is no need for brushes. The commutation is similar to the one in brushed DC motors, but it is performed by an electronic controller using a solid-state circuit rather than a commutator/brush system.

Compared with brushed DC motors, brushless motors have higher efficiency and reliability, reduced noise, longer lifetime due to the absence of brushes, elimination of ionizing sparks from the commutator, and reduction of electromagnetic interference. The stationary windings do not suffer with centrifugal forces. The maximum power that can be applied to a brushless DC motor is very high, limited almost exclusively by heat which can damage the permanent magnets. Their main disadvantage is higher cost which has been decreasing due to their mass production, as the number of applications involving them increases.

The better efficiency of brushless motors over brushed ones is mainly due to the absence of electrical and friction losses due to brushes. This enhanced efficiency of brushless motors is greatest under low mechanical loads and high speeds. However, high quality brushed motors are comparable in efficiency with brushless motors under high mechanical loads, where such losses are relatively small compared to the output torques.

Their kV rating is the constant relating the motor RPM at no-load to the supply voltage. For example, a 1,000 kV brushless motor supplied with 11.1 volts will run at a nominal 11,100 RPM.

Most brushless motors are of the inrunner or outrunner types. In the inrunner configuration, the permanent magnets are mounted on the spinning rotor in the motor core. Three stator windings are attached to the motor casing, surrounding the rotor and its permanent magnets. **Figure 1** shows a brushless inrunner of the KB45 series, used to power the spinning drum of our featherweight Touro Feather.

In the outrunner configuration, the windings are also stationary, but they form the core of the motor (as can be seen in the Turnigy motor in **Figure 2**) while the permanent magnets spin on an overhanging rotor (the "spinning can") which surrounds the core. Outrunners typically have more poles set up in triplets to maintain the three groups of windings, resulting in a higher torque and lower kV than inrunners. Outrunners usually allow direct drive without a gearbox because of their lower speed and higher torque. Due to their relatively large diameter, they're not a good option to be horizontally mounted inside very low profile robots.

Remember to leave a generous

FIGURE 1.

FIGURE 2.

FIGURE 3.

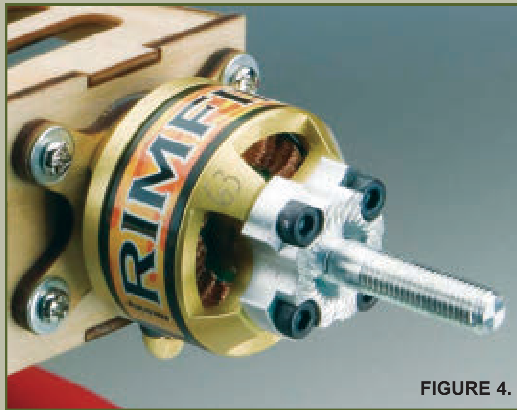


FIGURE 4.

FIGURE 5.

clearance all around an outrunner, to prevent its outer spinning can from touching any structural part of the robot that could be bent during combat.

Popular brushless outrunners are the ones from Turnigy and the more expensive ones from the famous Czech Republic company AXi, pictured in **Figure 3**. We've also tested very good outrunners from E-Flite (such as E-Flite's Park 250) and Little Screamers (such as the "De Novo" model).

One important thing about outrunners is that they should be mounted "behind the firewall" for combat applications. Firewall is the flat panel, cross-shaped mount, or standoff at the front of a model airplane where the motor is attached to. Supporting the motor in front of the firewall (as shown in **Figure 4**) is a good idea in model airplanes to help the motor cool down with the aid of the propeller air flow. The motor shaft mostly sees axial loads in this case.

Pulleys used to power robot weapons put large bending forces on the motor shaft. So, for combat applications it is important to support the motor by mounting it as close to the output shaft as possible, behind the firewall, as shown in **Figure 5**.

To mount outrunner motors behind the firewall, you might need to reverse the position of the output shaft for it to stick out from the face where the firewall is attached to. This can be done through the repositioning of the shaft retaining clips or screws.

Since most brushless speed controllers do not allow the motor to reverse its spin direction during combat, the use of brushless motors in combots is usually restricted to weapons that only spin in one direction. Reversible brushless speed controllers will soon become cheap and small enough to allow their widespread use in the robot drive system, as well.

More information on brushless motors can be found in the wikipedia link at **http://en.wikipedia.org/wiki/Brushless_DC_motor**. SV

MANUFACTURING: Hobbyweight Weapon Blade and Hub

● by Pete Smith

I needed a new blade for my 12 lb Hobbyweight combat robot Surgical Strike — an existing titanium blade with steel teeth was becoming increasing bent and if it lost a tooth in a fight, then it became so unbalanced as to be useless. I had the idea of combining an easy to manufacture 7075 aluminum hub with a watercut steel blade (**Figure 1**).

The hub has an accurately bored center that allows the use of a heavy duty keyless bushing (McMaster part 1058K13) and a 5/8" case hardened steel shaft (McMaster part 6061K111). A center boss locates the blade and takes much of the stress of any hits, and six 1/4-20 hex head screws attach the two parts together.

The blade is 14" long, 1.5" wide, and 1/4" thick. I had it cut from 4140 ChromeMoly steel and hardened to RC45 by Team Whyachi (**www.teamwhyachi.com**).

I had them make a pair; the cost was less than $150 each. If you wish

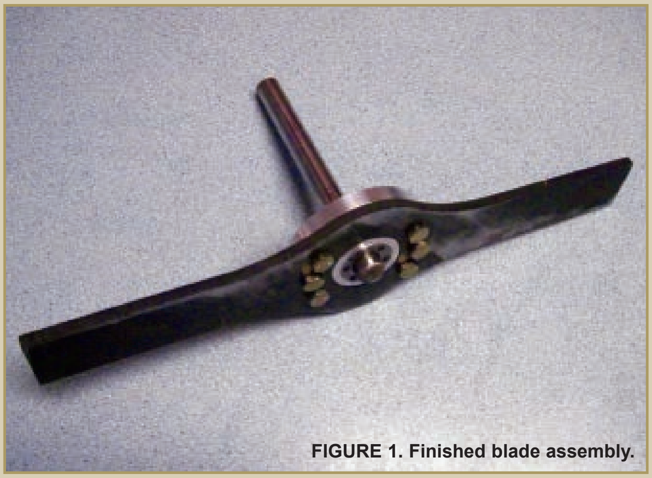


FIGURE 1. Finished blade assembly.

FIGURE 2. Face and turn to major diameter.

to make the blade yourself or have one made for you, then PDF, IGS, and DXF files are available in the article download on the *SERVO Magazine* website or through the Kitbots' website (**www. kitbots.com)**.

The hub is fairly easy to make on any lathe capable of taking a 3.5" diameter workpiece.

Drawings and an IGS file for the hub are also available in the downloads.

Safety Warning: Lathes are not toys and can easily injure or kill if used incorrectly. Read and follow all the manufacturer's safety notes. Do not wear loose clothing or jewelry, and always wear safety glasses.

I secured the 3.5" diameter bar in the three jaw chuck, faced off the end, and turned the exterior down to the required 3.25" (**Figure 2**). I was making two hubs, so I turned down a long enough section to allow for the two hubs plus the two cutoffs required to separate the parts from the original bar. I used 7075 aluminum as it is one of the toughest grades available, yet still machines easily.

The boss for the blade was turned down to the correct diameter (**Figure 3**) and I test-fitted the blade to the hub to check that it was a precision fit (**Figure 4**). If the fit is too loose, it will allow the blade to become unbalanced and also adds a lot more strain on the securing screws.

The next operation required the most accuracy. The bore for the keyless bushing has the tightest tolerance; this is required to ensure that the bushing grips the hub correctly. I first center-drilled (**Figure 5**), then drilled the part (**Figure 6**) with the biggest drill bit I had. I then used a boring bit (**Figure 7**) to increase the diameter to the required dimension. This requires patience (not really my forte) and frequent measuring. I bored the hole deep enough to allow for two hubs and the required cutoffs.

I then cut off the first hub and repeated the boss machining operation to make a second hub and cut that off, as well.

I used the blade as a pattern for positioning the mounting holes in the hub. I positioned the blade on the hub and used an appropriately sized transfer punch to locate one of the holes. I then drilled and tapped that hole and then repeated the operation for the remaining holes, adding each mounting screw in turn

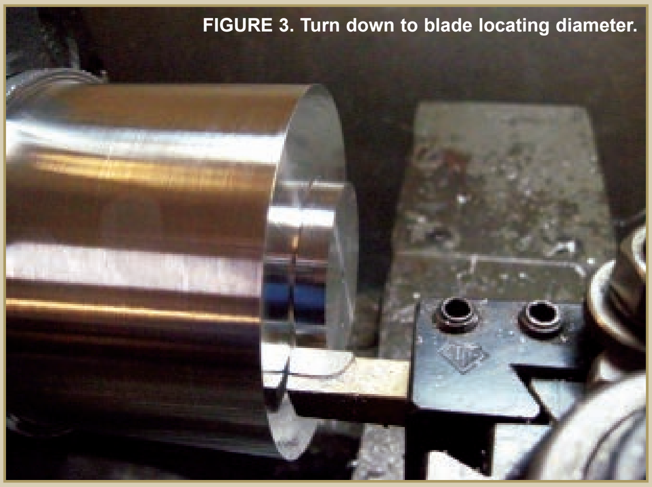


FIGURE 3. Turn down to blade locating diameter.

FIGURE 4. Trial fit blade to hub.

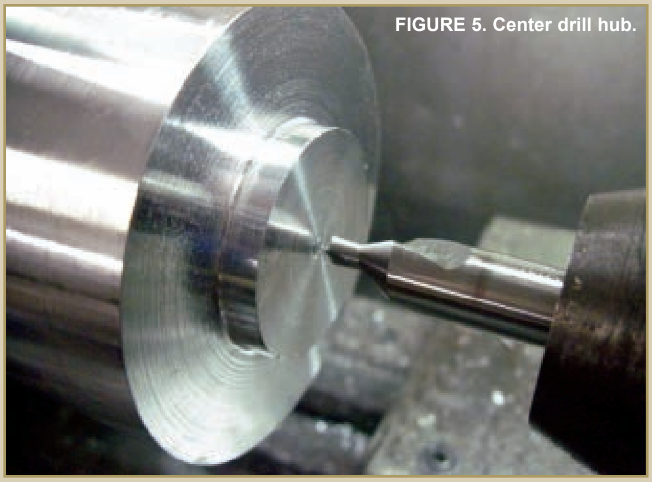


FIGURE 5. Center drill hub.

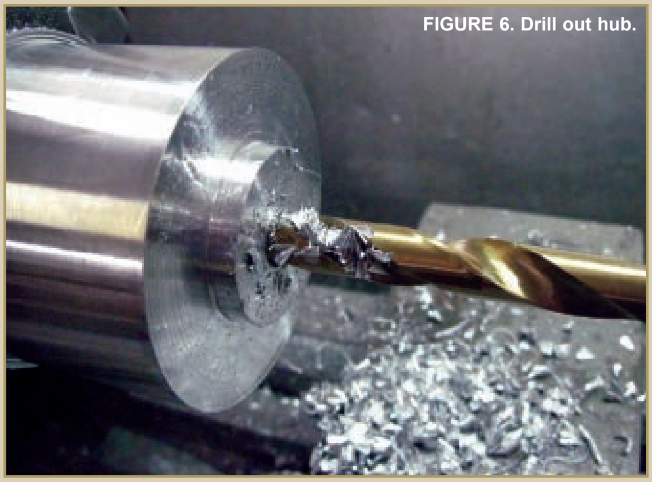


FIGURE 6. Drill out hub.

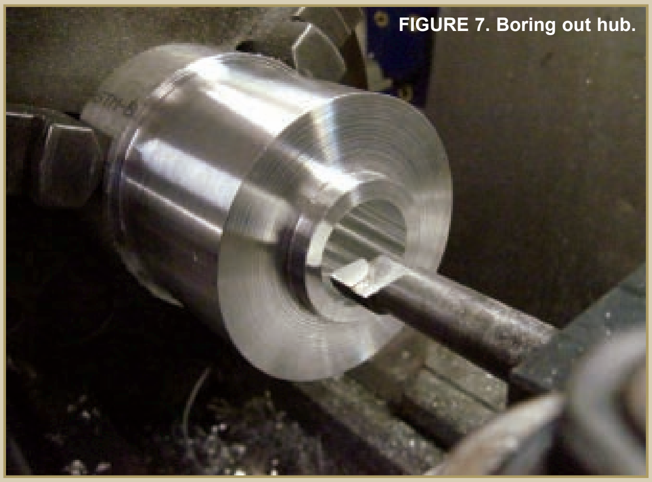


FIGURE 7. Boring out hub.

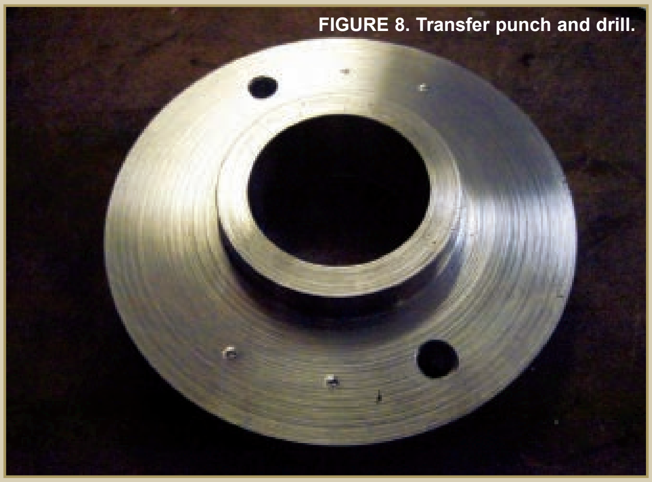


FIGURE 8. Transfer punch and drill.

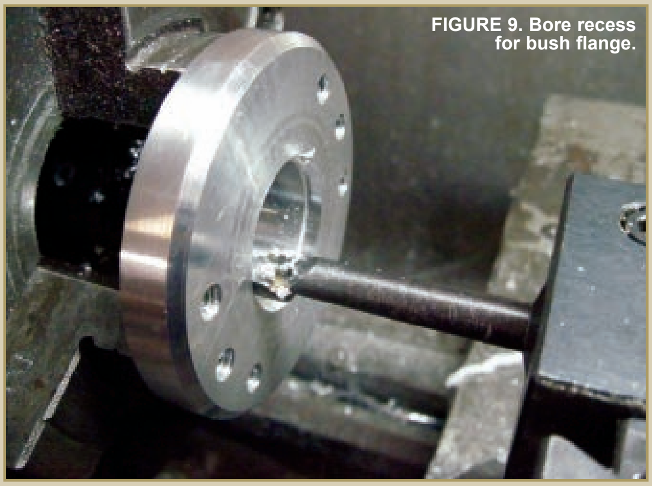


FIGURE 9. Bore recess for bush flange.


FIGURE 10. Fit blade to hub.

(**Figure 8**). This ensured that they all lined up correctly. One could use a rotary table or indexer to locate the holes, but the above method worked well enough for the small number of parts I was making.

The last machining operations were on the top side of the part. A recess was added to allow the flange of the bushing to sit flush with the top surface (this protects the expensive bushing and shortens the overall length of the assembly). This was done using the boring bar and is a simple clearance fit (**Figure 9**). I also added a small chamfer on the top outside edge to improve appearance and lose a little weight. It should be noted that neither of these operations required great accuracy or exact concentricity with the rest of the part. A three jaw chuck will produce very accurate work as long as the part is not taken

out of the chuck between operations. The part is remounted in the chuck using the blade boss for these topside operations, so strict accuracy is not maintained. If you do need high accuracy and have to move the part between operations, then a four jaw chuck and careful setup is required.

I fitted the blade to the hub using Grade 8 hex head screws (**Figure 10**). The hex heads are a little more likely to get damaged in combat, but will be much easier to replace if they do as you can usually get a pair of Vise-Grip pliers to get a solid grip on even a badly damaged head. Socket head screws often have the hex sockets damaged and can be a pain to get out in a hurry.

The new blade and boss were tested at the Franklin Museum event in Oct '10 and proved to work well. I would get the next set of blades a little harder perhaps, but it's always a balancing act between being too hard and breaking, and too soft and bending/blunting. I might also try a longer and thinner blade to give a little extra reach. SV

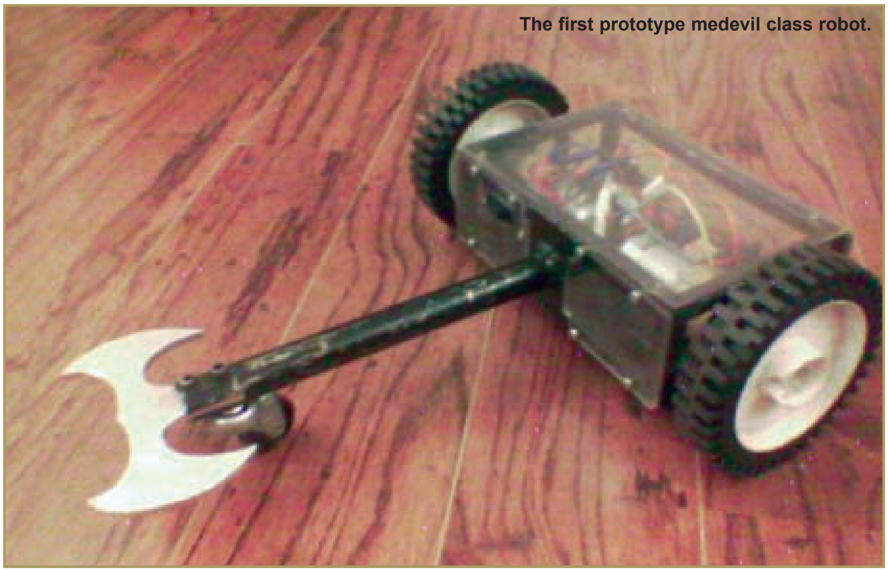
PARTS IS PARTS: Mtroniks Viper

● by James Baker

The Mtroniks Viper series of boat speed controllers have been used in many of my robots in the past. Made in Great Britain, these controllers offer full forward and reverse functions, one touch calibration, and advanced failsafe functions. They also have a built-in battery elimination circuit and motor stall protection.

There are a lot of speed controllers available for sub-featherweight robots (under 30 lbs) from lots of different manufacturers, and all have their advantages and disadvantages. For one specific application, I consider the Mtroniks Viper to be unrivaled. One of the areas of robot combat I am involved in is providing the opportunity for the paying public to come and drive identical robots in combat against each other. This "arrive and drive" attraction features robots specifically designed to fight only each other, in a quick assembly combat arena. This controlled combat format allows the robots to be optimized for long run times, quick battery changes, exciting battles, and reliable operation in any kind of weather. Yes, you read that right, these robots often run in the rain.

My smallest arrive and drive robots weigh under two lbs and use a pair of Viper15 speed controllers (which are 100% waterproof) coupled with silicone covered motors, and a single sealed 7.2 volt 3,000 mAh NiMH battery. The 15 amps rating of these controllers is more than enough to run these fast little machines without ever breaking a sweat — even in heavy rain. I have actually given demonstrations to potential customers where I have run one of these robots in a bath full of water. These robots actually float, but do regularly submerge completely when they crash into other objects. This rather extreme demonstration serves

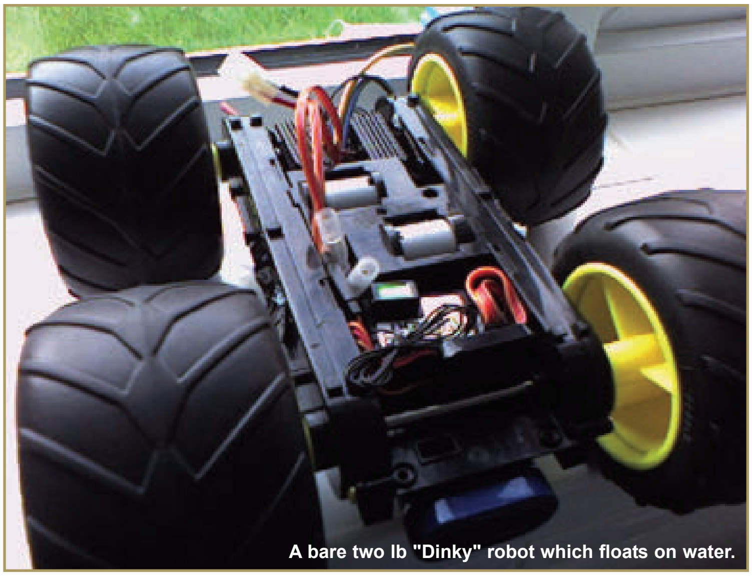


The first prototype medevil class robot.

Medevil in 2009, still with original controllers.

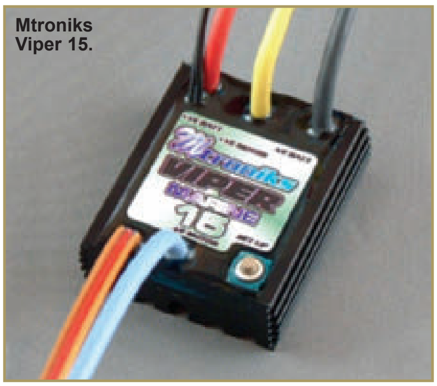
to illustrate the all-weather nature of the robots and the speed controllers, which is essential to many customers as the British weather can often interrupt outdoor events (and would otherwise close the attraction down until the rain stopped).

In my 12 lb "medevil" class of robots, I use the larger Viper 25 controller which is again over specification for the motors used. However, they give the reliability of having them running well under their maximum design loads. One of my prototype "medevil" robots is still in regular use as a marketing and display model, and it still uses the same two Viper speed controllers originally fitted in 2004. The robot itself is now looking rather worn, battered, and used (and is due for a complete rebuild), but the speed controllers look and function as new.



A bare two lb "Dinky" robot which floats on water.

An important thing to note at this point is that the second prototype "medevil" had a Viper controller fail for no apparent reason, over nine months after it was purchased. I phoned Mtroniks and sent the controller back to them, where it was replaced with a brand new unit within a few days and without any hassle at all. I've never known such rapid, stress free service from a supplier. The service element is at least as important as the actual product in my opinion, and Mtroniks definitely scored full marks there.



Mtroniks Viper 15.

Mtroniks Viper 25.

When I needed a brushed motor speed controller with true all-weather reliability, advanced failsafe features, great value for the money, and excellent customer service and support, then these were the things to have, so I recommend them whole-heartedly.

Visit **www.xbotz.com** or **www.mtroniks.net** for more information. **SV**

EVENTS

Upcoming Events for March – April 2011

Central Illinois Bot Brawl 2011 will be presented by the Central Illinois Robotics Club in Peoria, IL on April 2, 2011. If you would like further information, please go to **http://circ.mtco.com**.

RoboGames 2011 will be presented by ComBots in San Mateo, CA, on April 15th through 17th. If you would like further information, please go to **http://robogames.net/competing.php**. **SV**


It's okay. They do this every Thursday.
The SERVO readership post-retirement
$¢
2011


AndyMark
Inspiring Mobility
Specializing in Unique Wheels, Gearboxes, Aluminum Sprockets and Drive Bases
Tri- Lambda Drive Base
Omni-directional drive system kit.
6" Aluminum Dualie Omni Wheel
8" Mecanum Wheel
Aluminum Sprockets
AndyMark, Inc.
700 E. Firmin St., #114
Kokomo, IN 46902
765-868-4779
www.andymark.com


The Robot MarketPlace
Over 10,000 products available!
• Motors
• Batteries & Chargers
• Vex Robotics
• Carbon Fiber
• Wheels & Tires
• Speed Controllers
• R/C Systems
• Drive Components
• Hobby Supplies & Toys
(877) ROBOT 99
(877-762-6899)
Your One-Stop Robot Shop!
RobotMarketPlace.com

Harvest Energy For Your Robots with the EnerChip

by Fred Eady

The ancients invented today's battery. However, they didn't know that the sun they worshipped held the real key to portable electricity. Battery power is something humans use to enhance their daily lives. A robot sees a battery in a much different way. A battery may be a robot's life blood. Mobile robotic devices must either pack enough battery energy to complete their mission or have the ability to tap into energy sources that surround them.

This month's discussion revolves around energy harvesting. If the term "energy harvesting" is not in your everyday vocabulary, the Mars Spirit rover is a very good example of a robotic vehicle living "off of the land." The Spirit was only supposed to navigate the Martian surface for a total of 90 days. Six years later, the Spirit rover failed to communicate with its Earthly operators.

As of this writing, there are hopes that communications with the Spirit rover will be re-established. Meanwhile, the Opportunity rover — which is just a bit less than seven years old — is still clocking miles and harvesting Martian sunlight.

The EnerChip CBC050

Our energy harvesting discussion will revolve around the Cymbet EnerChip CBC050 — a 50 µAh rechargeable solid-state energy storage device. The EnerChip CBC050 is not just another rechargeable battery. It is a low profile, surface-mountable backup power source housed in an 8 mm x 8 mm QFN package. The CBC050 is designed to replace coin cells, super capacitors, and batteries that are normally used to provide backup power for embedded real time clocks, non-volatile memory, and microcontrollers.

Don't let the relatively tiny discharge rate of 50 µAh negatively influence your opinion of the CBC050's performance. With the right supporting components and some energy-aware host firmware, a pair of EnerChip CBC050's can pulse power embedded packet radios and the microcontrollers that host them using nothing more than ambient light.

The nominal output voltage of a CBC050 is 3.8 volts. The discharge cutoff voltage is 3.0 volts. If the discharge cutoff voltage is adhered to, the CBC050 can be recharged thousands of times. It is also able to retain a charge longer than standard rechargeable batteries. No special current limiting circuitry is required to charge one of these puppies. One needs only to provide a 4.1 volt charging voltage for 20 minutes to take a CBC050 energy storage device to 80% of capacity.

Fred Eady's *First Rule of Embedded Computing* — which states that "nothing is free" — applies to the EnerChip CBC050. As the number of charge cycles exceeds 1,000, the CBC050's internal resistance will increase. At room temperature (25° C), a new CBC050's internal resistance lies between 750 and 2,000 ohms. A typical CBC050 that has been recharged in excess of 1,000 times exhibits an internal resistance of 4,200 to 7,000 ohms. An increase in charge time can also be observed once the number of charge cycles exceeds 1,000. At 1,000 charge cycles, that 20 minute charge time to 80% of capacity becomes 60 minutes. However, if proper design techniques are followed, the load performance will be minimally compromised.

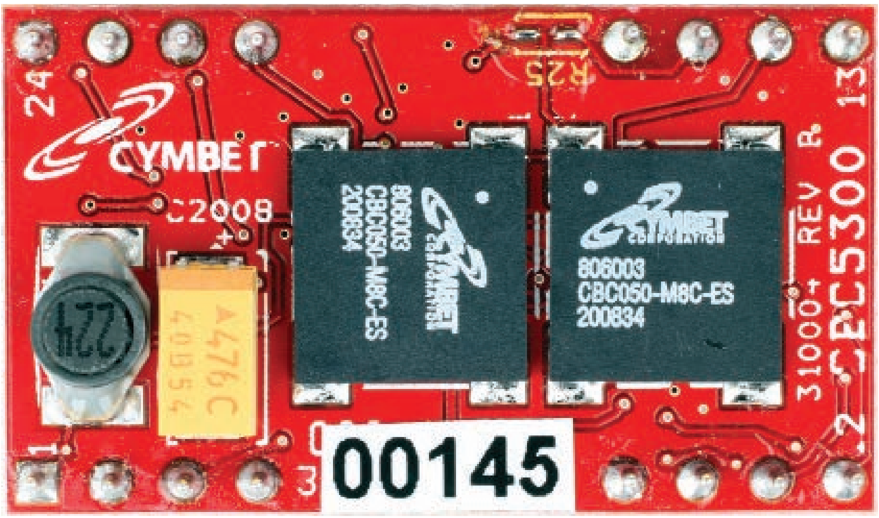


PHOTO 1. The Cymbet EnerChips shown here are 50 μAh devices tied in parallel. The capacitor and inductor to the left of the EnerChips are part of the voltage boost circuitry.

Solar Farming With the EnerChip

To harvest and utilize solar energy, we'll need some photovoltaic cells and a means of storing the converted electrical energy we collect. In an embedded environment, we may not have the luxury of cascading an array of multicelled photovoltaic devices to obtain a voltage that is large enough to directly charge a CBC050. So, we'll have to depend on some electronic magic to boost the voltage supplied by the photovoltaic cells to a level that the CBC050 can chew on.

The energy harvesting electronic assembly gleaming in **Photo 1** is officially known as a CBC5300 EnerChip EH Energy Harvesting Module. The CBC050 rechargeable solid-state energy storage devices riding on this module are rated at 50 μAh each and are electrically paralleled to provide a combined 100 μAh of available energy discharge.

Photo 2 is the B side of the EH Module. The electronics on side B form a number of power management and power control subsystems. Basically, the CBC5300 Module accepts input from an energy harvesting transducer which transforms ambient energy into electrical energy. The energy harvesting transducer can be photovoltaic, piezoelectric, or thermoelectric. The incoming electrical energy gathered by the input transducer is normally insufficient to charge EnerChip solid-state storage devices. Thus, the first power management subsystem the incoming electrical energy sees is a boost converter which amplifies the incoming voltage to a level that can charge the EnerChip storage devices and drive the power management circuitry.

A charge control subsystem continuously monitors the output of the boost converter. If the boost converter output voltage isn't sufficient enough to charge the EnerChips, the charge control subsystem disconnects the EnerChips from the boost converter. Without this ability to disconnect, the EnerChips would attempt to back-power the boost converter. Discharging the EnerChips below the 3.0 volt minimum discharge voltage will damage these energy storage devices. Discharging below 3.3 volts is not recommended.

A power management subsystem monitors the output of the EnerChips. The purpose of the power management subsystem is to prevent the EnerChips from discharging too deeply. In addition to providing a smooth power-on transition to the load, the power management subsystem provides an active low CHARGE control line that indicates when the energy harvesting electronics are actually charging the EnerChips.

Walking Around the CBC5300 EnerChip EH Energy Harvesting Module

Our Module checklist is in the form of **Figure 1**.

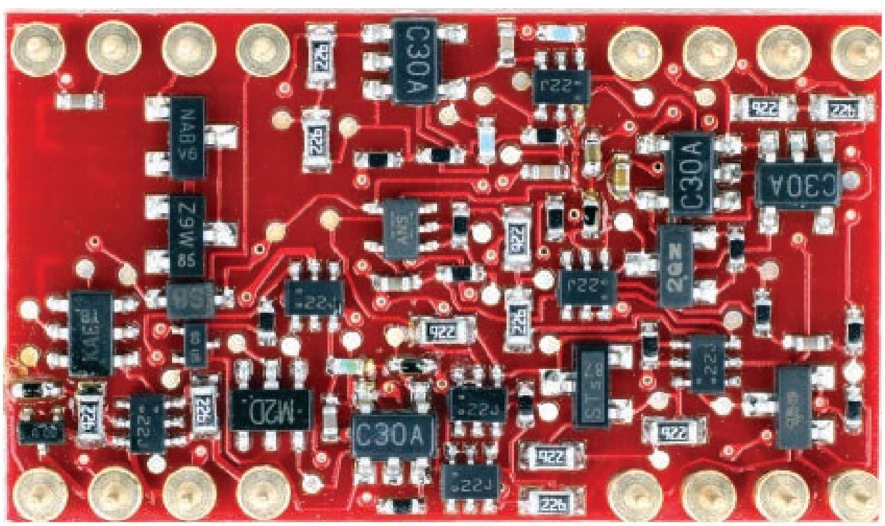


PHOTO 2. There's no need to attempt to reverse-engineer this circuit. Trust me. It's a voltage boost converter and power management system for the EnerChips.

Pin Number	Label	Description
1	V_{IN}	Transducer power input
2	PGND	Same potential as system ground. Tie to GND.
3	GATE	Output signal used for setting operating point. Leave unconnected or use as a test point.
4	$\overline{\text{CHARGE}}$	Active low output from the CBC5300 indicating that the EnerChips have been charged or are being charged. This is an open drain output with an internal 10MΩ pull-up resistor to V_{OUT}. See the section Circuit Recommendations to Save Power for additional information.
5	NPP	No pin present
6	NPP	No pin present
7	NPP	No pin present
8	NPP	No pin present
9	SIGRTN	Tied to bottom of voltage divider to set V_{OPER}
10	V_{OPER}	Set peak power operating voltage
11	V_{REG}	Tied to top of voltage divider to set V_{OPER}
12	UVLO SEL	Under voltage lockout select. The nominal set point is 700 mV. Normally, this pin is tied to V_{IN}. To raise the threshold, add one or more series diodes between this pin and V_{IN}.
13	EC2GND	Factory test pin - tie to pins 14, 2 and 24 on target
14	EC1GND	Factory test pin - tie to pins 13, 2 and 24 on target
15	BATTEST	Factory test pin - no connection
16	BATOFF	Input control line to the CBC5300 for disconnecting the EnerChips from the CBC5300 charging circuit. See the section Circuit Recommendations to Save Power for additional information.
17	NPP	No pin present
18	NPP	No pin present
19	NPP	No pin present
20	NPP	No pin present
21	V_{CAP}	Output holding capacitor interface
22	V_{BAT}	EnerChip power output, connected indirectly to the EnerChips' positive terminals through an isolation FET. Voltage is one diode drop above the potential at V_{OUT}.
23	V_{OUT}	System output power (3.6V maximum)
24	GND	System ground

FIGURE 1. There aren't very many knobs to tweak but every knob has a great deal to do with utilizing the harvested energy as efficiently as possible.

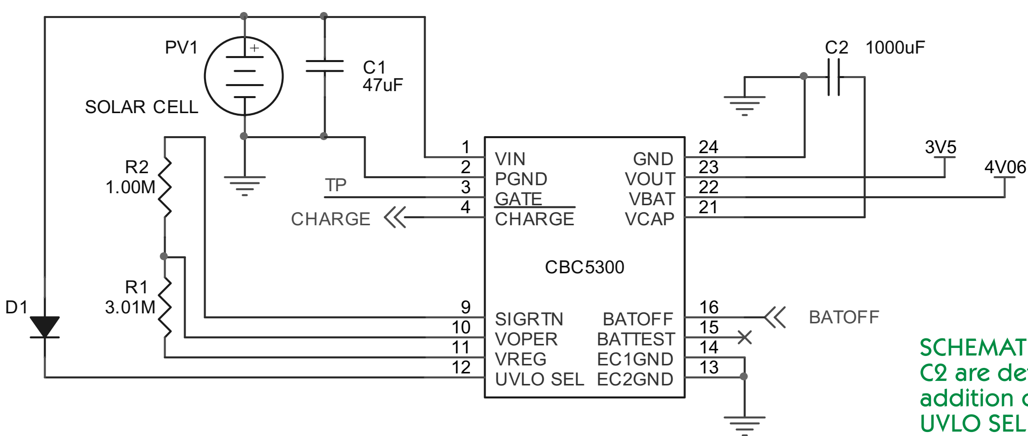
Although the CBC5300's V_{IN} input can handle up to 4.0 volts, the most efficient voltage to present at V_{IN} is 0.9 volts, and although the maximum voltage that can be safely applied to V_{IN} is 4.0 volts, the CBC050 datasheet recommends that no more than 1.5 volts be applied to V_{IN}. When used as suggested in the CBC5300 datasheet, the EH Module produces a nominal +3.55 VDC output. If you're thinking about using V_{BAT} to power portions of your target circuit, that's a NO NO as V_{BAT} is reserved for factory test operations.

The charge control system's disconnect feature can be emulated by the host microcontroller using the Module's BATOFF control line. BATOFF is used to halt EnerChip charging and direct the full power of the EnerChips to the output load. Thus, when BATOFF is driven logically low, the charge control subsystem operates normally and the EnerChips will always be charging when the input power is available. The only other direct connection to the host microcontroller's I/O subsystem is made at the Module's CHARGE pin. This pin acts as an output signal to inform the host that the EnerChips have been fully charged. The CHARGE pin also enters an active-low state when the transducer output power is capable of driving the boost converter.

Integrating the CBC5300 EnerChip EH Energy Harvesting Module

There are many robotic applications in which our Module can participate. One of the most useful robotic applications involves communications. Recall that I mentioned earlier that a pair of CBC050s with the right supporting cast could easily drive an embedded packet radio. Naturally, the radio would most likely be used to communicate the condition of an attached sensor. With that, let's see what it takes to adapt the EH Module to that purpose.

A typical CBC5300 Module application circuit is depicted graphically in **Schematic 1**. Upon seeing this schematic, the first thing I asked myself was how they came



SCHEMATIC 1. The values of R1, R2, and C2 are determined by formulae. The addition of diode D1 raises the 700 mV UVLO SEL voltage by a diode drop to 1.4 volts.

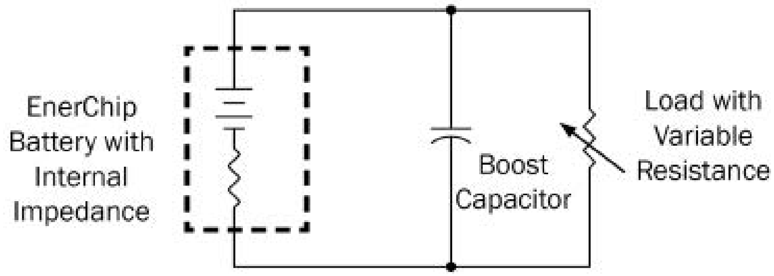


FIGURE 2. This scheme works because the ESR of the boost capacitor is very low which allows the combined current of the EnerChip and boost capacitor to flow freely to the resistive load.

about the resistor and capacitor values. The answer to that question lies in the understanding of why the components are there.

The Microchip MRF24J40MA RF Transceiver module draws an average of 23 mA during transmit and 19 mA in receive mode. Those current drain figures are pretty much standard across the board as far as RF transceivers that fall into the same device class as the MRF24J40MA. The obvious question at hand is how can the 100 µAh pair of EnerChip CBC050 energy storage devices supply enough power to drive the MRF24J40MA? Part of the answer lies in that low ESR (Effective Series Resistance) 1,000 µF capacitor tied across the EH Module's V_{CAP} terminal and ground. Capacitor C2 is sized to be a current reservoir that is capable of pulsing enough supplemental current to satisfy the attached load. Thus, between the more current-capable EnerChip and the relatively lower-charged low ESR output capacitor, steady state power is presented to the load. The EnerChips charge the output boost capacitor between pulses. This concept is illustrated in **Figure 2**.

The amount of pulse current energy that is delivered depends on a multitude of battery factors:

- Battery Impedance
- Battery Voltage
- Operating Temperature
- Pulse Current Amplitude
- Pulse Current Duration

If we know the pulse current amplitude and pulse current duration, we are on the way to being able to size the boost capacitor (C2). Since the EnerChip has to recharge the boost capacitor and that recharge process takes a finite amount of time, we need to know what that finite amount of time needs to be.

A MRF24J40MA RF transceiver in a wireless sensor node application will draw a minimum of 23 mA. The transmit duration in such an application can vary between five and 50 mS. A typical wireless sensor node transmission is around 20 mS. We want to provide enough time between pulses to recharge the boost capacitor while not allowing the load voltage to fall below 3.0 volts or the EnerChip charging voltage to drop below 4.1 volts.

The MRF24J40MA can operate with a voltage as low as 2.4 volts. So, let's set our minimum MRF24J40MA operating voltage at 3.0 volts. Each 20 mS transmission will draw a maximum of 23 mA.

EnerChip internal impedance will increase as the state of charge decreases. If we use the worst case impedance number at the worst case temperature with a 10% state of charge, the EnerChip internal impedance will be around 7 KΩ. I didn't pull those figures out of my hat. You can come to the same conclusion by using the charts in the Cymbet application note AN-1025. In AN-1025, you will also find that at 10% of charge at 25° C, the EnerChip voltage will be a fraction over 3.8 volts. If you're wondering why we're producing all of these numbers, we're going to use them to compute the size of boost capacitor C2. Here's the formula as supplied in AN-1025:

$$C = t/\{R * [-\ln (V_{MIN} / V_{MAX})]\}$$

Where:
C = Output capacitance (in parallel with battery)
t = Pulse duration
R = Load resistance = V_{OUT}(average) / Ipulse

Here's what we know right now:

t = 20 mS
V_{MIN} = 3.0 volts
V_{MAX} = 3.8 volts

We still need R. So, let's compute it. Believe it or not, R is derived from Ohm's Law as the average voltage divided by the load current:

$$R = [(3.8 + 3.0) / 2] / 23 \text{ mA} = 148 \text{ Ohms}$$

Now, we can solve for C:

$$C = 20 \text{ mS} / \{148 * [-\ln (3.0/3.8)]\} = 572 \text{ µF}$$

The next higher standard capacitor value is 680 µF.

Okay. We have the boost capacitor value. From here, it's a walk in the park to calculate the time needed between pulses to make sure that the boost capacitor is fully charged before the next transmit pulse occurs. Here's the formula according to AN-1025:

$$t = R * C * [-\ln (1 - V_{MIN} / V_{MAX})]$$

Where:
t = Capacitor charging time (V_{MIN} to V_{MAX})
R = Battery resistance
C = Output capacitance (in parallel with battery)

PHOTO 3. This energy harvesting module is made up of photovoltaic cells coupled with a pair of paralleled rechargeable EnerChip solid-state energy storage devices.

Let's take that walk in the park:

$$t = 7{,}000 \text{ ohms} * 680\ \mu F * [-\ln(1 - 3.0 / 3.8)] = 7.4 \text{ seconds}$$

Adding some insurance time brings us to a 10 second transmit interval between pulses.

If we were to work backwards to match the 1,000 µF boost capacitor in **Schematic 1**, using the same formulae criteria would result in an available pulse duration of just under 40 mS and a calculated transmit interval of 11 seconds.

With the boost capacitor value determined, let's move on and calculate the V_{OPER} value. V_{OPER} is a datasheet value that is particular to the transducer. So, we're going to work backwards to calculate the V_{OPER} value using the R1 and R2 resistor values noted in **Schematic 1**. Here's the formula to calculate V_{OPER}:

$$V_{OPER} = V_{REG} * (R2 / (R1 + R2))$$

Where:
V_{REG} = 4.06 volts

To calculate V_{OPER}, we simply employ substitution:

$$V_{OPER} = 4.06 * (1.00M / (3.01M + 1.00M)) = 1.01 \text{ volts}$$

We can check our work using the formulae to calculate the value of R1:

$$R1 = R2 * ((V_{REG} / V_{OPER}) - 1) = 3.019\ M\Omega$$

The R2 resistor value can range from 500 KΩ to 1 MΩ, and is optimal at 750 KΩ. The standard 1 MΩ value was chosen because the target V_{OPER} voltage is farther away with the 750 KΩ value. The nearest standard value of 3.01 MΩ was also chosen for R1.

Believe it or not, diode D1 is also an "adjustment" component. The UVLO SEL (Under Voltage Lockout) voltage should be set above the voltage present at V_{OPER}. From our calculations, we know our V_{OPER} voltage is 1.01 volts. Without D1, the UVLO voltage is set at 700 mV. One diode drop is also about 700 mV. So, by inserting the diode between V_{IN} and UVLO SEL, we add a diode drop to the nominal 700 mV UVLO voltage which raises the UVLO voltage to 1.4 volts. Other desired UVLO voltages can be obtained by mixing in diodes with differing voltage drops such as Schottky diodes.

Touchy Feely Time

The hardware behind the energy harvesting schematic and math is posing for the camera in **Photo 3**. The EnerChip EH Solar Energy Harvester Evaluation Kit (CBC-EVAL-08) is the companion energy harvesting platform of the Microchip XLP 16-Bit Energy Harvesting Development Board. All of the energy harvesting electronic components and modules we've discussed (including the solar array) are loaded on the CBC-EVAL-08.

Now that you have more than a working knowledge of how energy harvesting works, you can apply what you've learned here to your own low-power robotic projects. If you're one of those folks that used to pull the battery packs out of Polaroid Instant film cameras, energy harvesting is in your future. **SV**

Fred Eady can be reached via email at fred@edtp.com

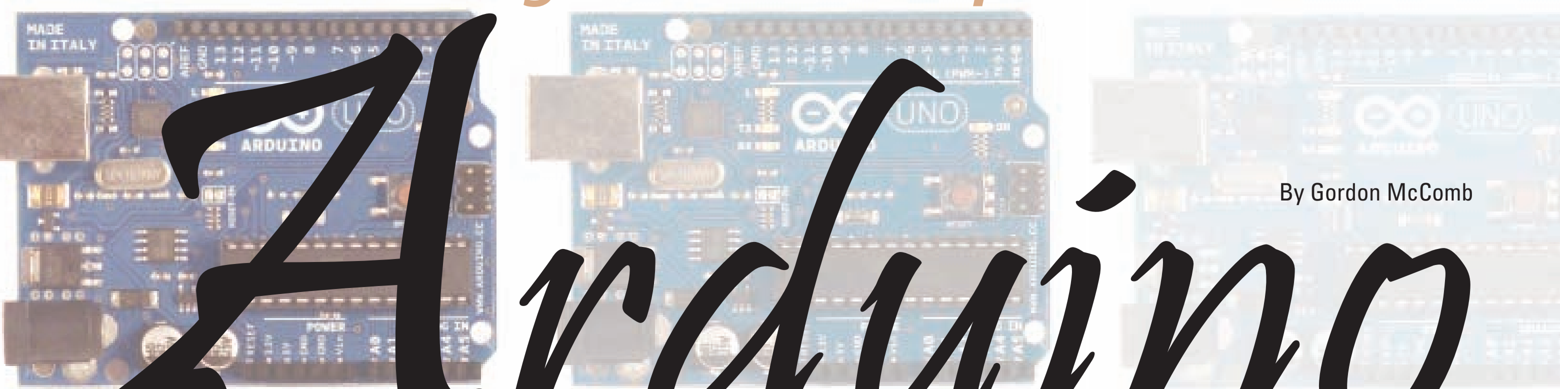
Sources

Microchip
XLP 16-Bit Energy Harvesting Development Board
MRF24J40MA
www.microchip.com

Cymbet
CBC5300 EnerChip EH Energy Harvesting Module
EnerChip CBC050
www.cymbet.com

Making Robots With The Arduino

Part 5 – Adding Sensors To Explore The World



By Gordon McComb

You've spent hours designing and building your latest robot creation. You bring it into your living room, fire it up, and step back. Behold! Your beautiful new robot smashes into the fireplace and scatters itself into tiny pieces over the living room rug. You remembered things like motor speed controls, colorful blinky lights, even a synthetic voice, but you forgot to provide your robot with the ability to look before it leaps.

In last month's installment, you learned how to give your ArdBot (or other Arduino-based automaton) the sense of touch and light. Augmenting these basic senses are methods to detect objects in *proximity* to your robot — seeing what's there without having to actually bump against them.

Proximity detection forms the basis of *collision avoidance* — how to keep your bot from crashing into things in the first place. Collision avoidance takes many forms. Some of these techniques are designed to detect objects very close to the robot, while others are made to detect objects several feet away.

In this installment, you'll learn about two popular forms of proximity detection — ultrasound and infrared — and how these low cost sensors are interfaced and used with the Arduino.

The State of the ArdBot

This article series has been on the construction and use of the *ArdBot* (see **Figure 1**) — an inexpensive two-wheeled differentially-steered robot based on the popular Arduino Uno and compatible microcontrollers. Here's what past installments covered:

- Part 1 (Nov '10) introduced the ArdBot project, along

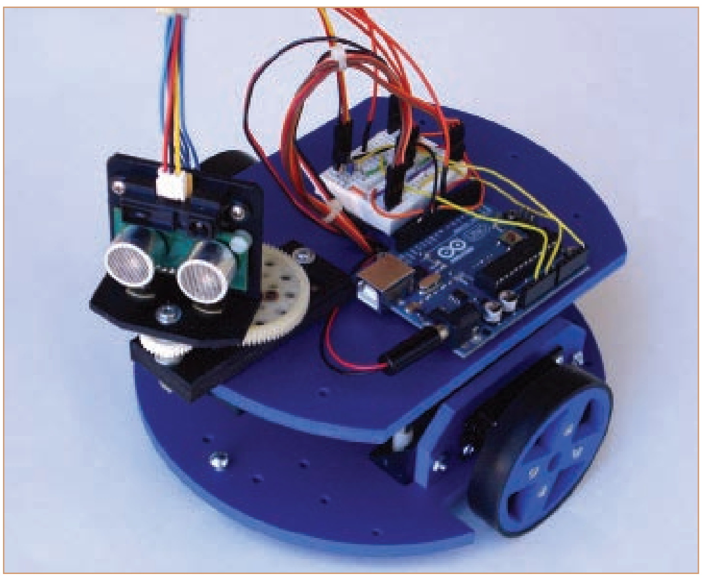


FIGURE 1. The ArdBot Arduino-based expandable robot, shown with this month's enhancements: a rotating turret, ultrasonic distance ranger, and infrared proximity detector.

Where Have All the Sharp Distance Sensors Gone?

From time to time, Sharp discontinues some models of their infrared distance sensors, allowing the stock that remains "in the channel" (still held in reseller inventory) to be depleted. New versions are in production, but they may not be universally available.

You may have noticed that several of the Sharp infrared distance sensors have become hard to get from some sources — most particularly from Digi-Key, who at one time was a major seller of the things. These sensors are discontinued and/or not available as new stock in retail channels.

However, most of the sensors can still be purchased from a number of specialty online shops, such as SparkFun and Pololu, both of whom carry large inventories of several Sharp models. What remains of the current stock may not be useful for creating a new mass-market product, but there's plenty still around for us robo-tinkerers.

Sharp has introduced a relatively new style of sensor — the GP2Y0D805 and GP2Y0D810 — that are smaller and less expensive. The '05 has a set 5 cm proximity range; the '10 has a set 10 cm range. Output is a digital LOW or HIGH.

This new class of sensor comes in a DIP-size package, but it requires some external components. Online retailers such as Pololu (see the **Sources** box) offer the sensors on a breakout board for easy use in your projects. Even with the addition of the breakout board, these sensors are roughly half the cost of their predecessors. They also have a much improved response time: over 350 Hz (350 updates per second), as opposed to about 25 Hz of the older sensors.

with the popular Arduino microcontroller and basic programming fundamentals of this powerful controller.

- Part 2 (Dec '10) detailed the construction of the ArdBot, using common materials including PVC plastic and aircraft grade plywood.
- Part 3 (Jan '11) covered the Arduino in more detail. It also examined the ins and outs of programming R/C servo motors that provide the locomotion for the ArdBot.
- In Part 4 (Feb '11), we learned about using bumper switches with polling and hardware interrupts to detect physical contact with objects, and ways to use photoresistors to steer the robot by light.

Be sure to check out the previous four issues of *SERVO Magazine* for more details. This installment describes three important robotic functions: programming an ultrasonic transducer to accurately measure the distance to objects; how to use a Sharp infrared distance module for monitoring proximity; and ways to add a rotating turret so the ArdBot can scan the room to look for things nearby.

In the next issue, we'll conclude with putting all the pieces together, combining what you've learned to create an autonomous room wanderer that's able to seek things out, investigate its environment, and follow the direction of its master — that's you.

Sources

If you'd like to build the ArdBot, be sure to start with the November '10 issue of *SERVO Magazine* for Part 1 of this series. Also check out the following sources for parts:

Prefabricated ArdBot body pieces with all construction hardware; 360 degree rotation sensor turret:

Budget Robotics
www.budgetrobotics.com

Arduino Resources:

Arduino
www.arduino.cc

Fritzing
www.fritzing.org

Online Retailers of Arduino, Sharp sensors, and/or ultrasonic sensors:

Acroname
www.acroname.com

AdaFruit Industries
www.adafruit.com

HVW Tech
www.hvwtech.com

Jameco
www.jameco.com

Lynxmotion
www.lynxmotion.com

Mark III Robot Store
www.junun.org

Pololu Robotics & Electronics
www.pololu.com

Robotshop
www.robotshop.com

SparkFun
www.sparkfun.com

About Non–Contact Near–Object Detection

Last time, we learned how leaf switches are used as a form of touch sensor to detect contact with objects. Contact detection provides an immediate signal that something looms directly in the way.

Non-contact detection senses objects without having to hit them first. *Near-object detection* does just what its name implies: It senses objects that are close by, from perhaps just a breath away to as much as eight or 10 feet. These are objects that a robot can consider to be in its immediate space; objects it may have to deal with, and soon. These objects may be people, animals, furniture, or other robots.

By detecting them, your robot can take appropriate action which is defined by the programming you give it. Your bot may be programmed to come up to people and ask for their name. Or, it might be programmed to run away whenever it sees movement. In either case, it won't be able to accomplish either behavior unless it can detect the objects in its neighborhood.

There are two common methods of achieving near-object detection: proximity and distance.

- Proximity sensors care only that some object is within a zone of relevance. That

is, if an object is near enough to be considered important. Objects beyond the proximal range of a sensor are effectively ignored because they are not seen. Out of view, out of mind.

- Distance measurement sensors determine the space between the sensor and whatever object is within detection range. Distance measurement techniques vary; almost all have notable minimum and maximum ranges. Few yield accurate data if an object is smack-dab next to the robot, or very far away.

Collectively, these sensor types are often referred to as rangefinders, though only a device that actually measures and reports the distance of the covered range is a true rangefinder.

Among the most common proximity and distance measurement detectors used in robotics are ultrasonic transducers, and specialty infrared sensors made by Sharp. Depending on the design of the specific sensor, either can be used for proximity or distance measurement. In practice however, the Sharp IR sensors are best suited for proximity, and ultrasound sensors are the ideal choice for measuring distance. That's how these two detectors are used in the ArdBot.

Using the Sharp GP2D120 Infrared Detector

The Sharp GP2D120 is among a series of remarkable distance measuring devices originally intended for industrial control. They're common finds in amateur robot projects. These sensors rely on the displacement of reflected light across a linear sensor (see **Figure 2**).

Here's how they work: A beam of modulated infrared light from the sensor illuminates an object. The beam reflects off the object and bounces back into the sensor. The reflected beam is focused onto what's known as a position sensitive device, or PSD. The PSD has a surface whose resistance changes depending on where light strikes it. As the distance between sensor and object changes, so does the linear position of the light falling on the PSD. Circuitry in the sensor monitors the resistance of the PSD element, and calculates the distance based on this resistance.

Among the Sharp detectors currently available through retail channels are two general types:

- Distance judgment sensors provide a simple digital (LOW/HIGH) signal that represents whether an object is within detection range. That range is set at the factory, and depends on the specific model of the sensor. Common distances are five, 10, 24, and 80 centimeters.

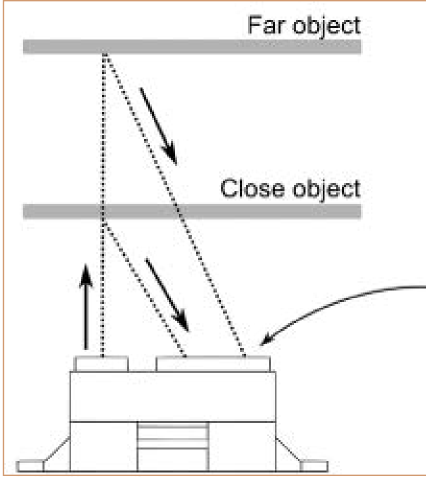


FIGURE 2. The Sharp infrared distance modules rely on the displacement of reflected light across a linear plane to detect variations in distance.

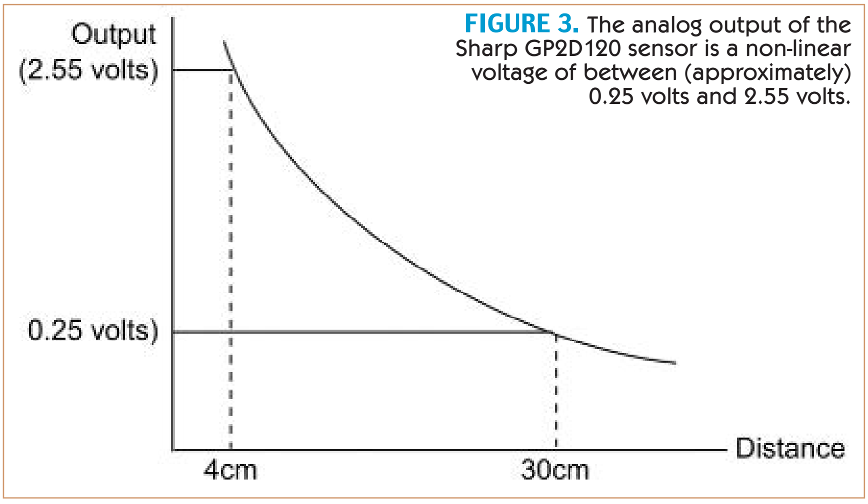


FIGURE 3. The analog output of the Sharp GP2D120 sensor is a non-linear voltage of between (approximately) 0.25 volts and 2.55 volts.

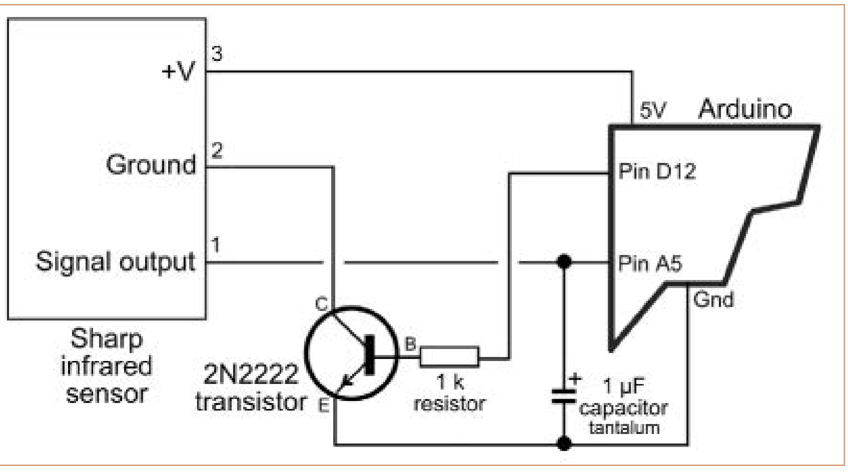


FIGURE 4. Connection schematic for the Sharp GP2D120, showing a transistor used to turn the sensor on and off.

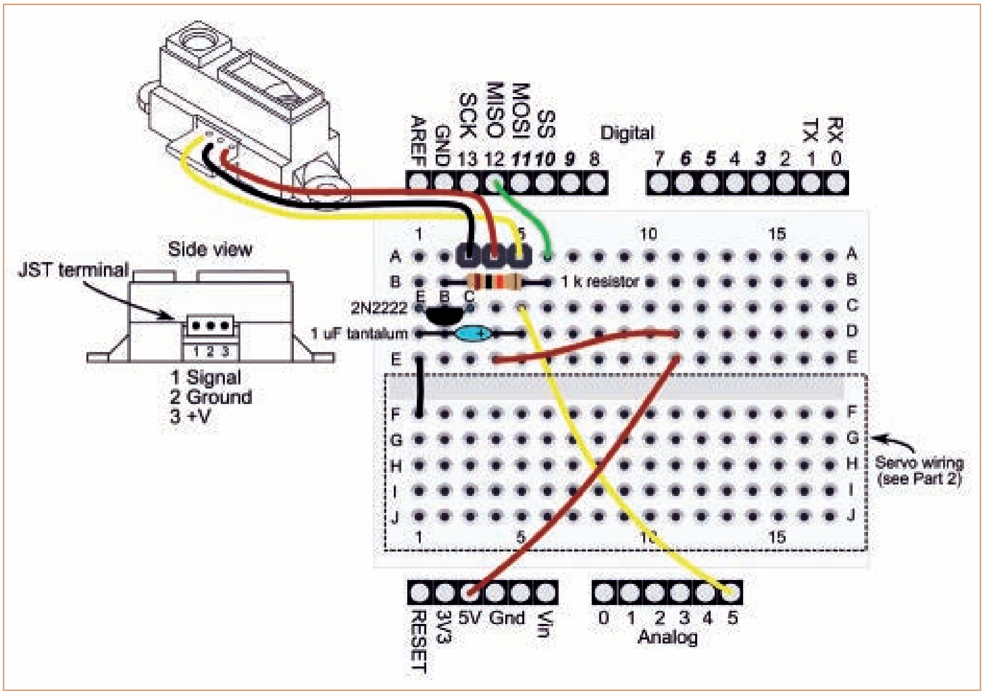


FIGURE 5. Breadboard view of the Sharp GP2D120-to-Arduino connection. Note the pinout and wiring order of the GP2D120 sensor. Important! The wiring diagram relies on existing connections on the breadboard. See Part 2 of *Making Robots with the Arduino* (Dec '10) for details.

For example, with the GP2Y0D810 sensor, the output is a digital LOW when an object is within its 10 cm proximity range, and HIGH otherwise.

- Distance measurement sensors provide an analog voltage that's more or less proportional to the distance from the sensor to the detected object. The voltage output is non-linear, as shown in **Figure 3**. These detectors work over a span of minimum and maximum distance, usually no closer than a few inches and up to several feet.

Distance judgment sensors are ideal for interfacing with simple electronics, as they don't require analog-to-digital conversion. Since the Arduino Uno is equipped with a six-input ADC, we can use either type. For the ArdBot project, I've selected a GP2D120 which has a range of 4 cm to 30 cm (about 1.5 inches to 12 inches).

The distance is reported as a varying voltage, from approximately 0.25 volts (no detection) to 2.55 volts (detection at minimum distance). That's according to the spec sheet, but know that there can be a normal variation of a few tenths of a volt from one sensor to another.

While the GP2D120 is capable of reporting distance with acceptable accuracy, for the ArdBot I've elected to use it as a "multi-zone" proximity detector. That is, instead of hassling with converting its analog voltage to some quasi-precise distance measurement, for the ArdBot the GP2D120 will instead simply indicate when an object is within preset zones.

The ArdBot relies on a separate ultrasonic rangefinder for accurate distance measuring and as a secondary near-object detection check. More about the ultrasound rangefinder in a bit.

See **Figure 4** for a schematic diagram for connecting the GP2D120 to the Arduino. **Figure 5** has the same circuit, but in breadboard view. See that I'm being clever here, and I've added a small 2N2222 NPN type transistor in order to turn the GP2D120 on and off.

The transistor is an optional enhancement, but there are a couple of reasons for it. First, like all of the Sharp sensors, the GP2D120 takes constant measurements — about 25 a second — as long as the device is powered. Current consumption can go as high as 50 milliamps when no object is detected. While that's not a huge current demand, it's unnecessary power consumption when the sensor is not actually being used.

Listing 1 – SharpIR.pde.

```
const int irCtrl = 12;       // Digital pin D12
const int irSense = A5;      // Analog pin A5
int distance = 0;

void setup() {
  pinMode(irCtrl, OUTPUT);
  digitalWrite(irCtrl, LOW);
  Serial.begin(9600);        // Use Serial Monitor window
}

void loop() {
  Serial.println(irRead(), DEC);
}

int irRead() {
  int averaging = 0;

  // turn on, wait 250 ms to completely stabilize
  digitalWrite(irCtrl, HIGH);
  delay(250);

  // Get a sampling of 5 readings from sensor
  for (int i=0; i<5; i++) {
    distance = analogRead(irSense);
    averaging = averaging + distance;
    delay(55);               // Wait 55 ms between each read
  }
  distance = averaging / 5;  // Average out readings
  digitalWrite(irCtrl, LOW); // Turn sensor off
  return(distance);
}
```

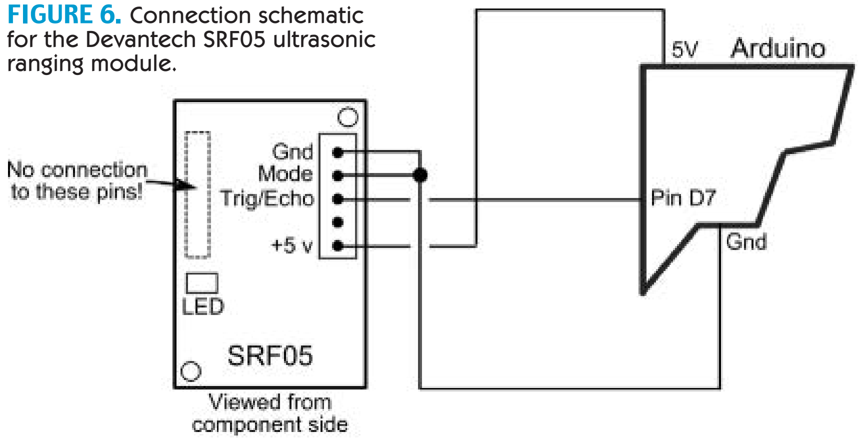
Second, each time the sensor takes a new reading there's extra line noise induced into your robot's power supply. By turning the sensor off when it's not needed, the noise is completely removed. (You can also help filter the noise by adding some decoupling capacitors across the +V and ground power connections, as close to the sensor as possible. Try a 47 µF electrolytic capacitor and a .1 µF ceramic capacitor. Be sure to observe correct polarity of the electrolytic capacitor.)

Listing 1 shows a sketch that demonstrates how to use the GP2D120 with the Arduino. This is an example sketch only; in the next installment, you'll see how the GP2D120 can be implemented for object seeking and avoidance, as the ArdBot is set loose in your living room and left to discover what's around it.

A couple of things to note in this sketch:

- To make a reading, the sensor is turned on and then allowed to settle for 250 milliseconds (ms) before taking a reading. The datasheet for the GP2D120 indicates a much shorter delay upon startup, but I've found the longer period is often necessary to avoid spurious reads.
- The sketch takes five "samples," each separated by a delay of 55 ms. The five samples are averaged to remove possible incorrect readings due to momentary glitches.
- After the sensor is read, it's turned back off again.

FIGURE 6. Connection schematic for the Devantech SRF05 ultrasonic ranging module.



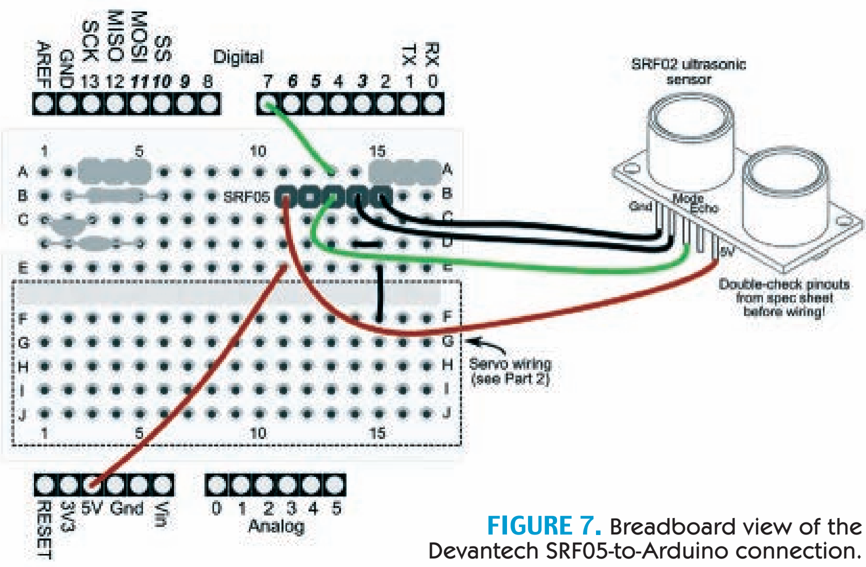


FIGURE 7. Breadboard view of the Devantech SRF05-to-Arduino connection.

Open the serial monitor window to observe the actual values reported by the sensor, converted from an analog voltage to 10-bit digital values (0 to 1023) by the Arduino's ADC. Note that because the sensor does not output a full five volts when an object is closest, you won't get the full 1,024 steps. Minimum values are about 40 to 60; maximum values are in the 625 to 675 range, depending on the sensor and the reflectivity of the object (dark colors tend to produce slightly lower values). Anything outside these ranges can indicate a spurious reading.

Important! Take note of the connection diagram for the GP2D120. Sharp uses a polarized JST connector where the +V power lead is on the outside and ground is in the middle. This is potentially dangerous when the other end of the wiring is non-polarized, as is often the case with connectors on general-purpose microcontrollers.

Numerous sources (such as Lynxmotion) sell adapter cables that go from the JST locking connector to a standard three-pin 0.100" female header. On many of these cables, the wiring order is "corrected" to place the +V power lead in the center. In this arrangement, damage is less likely to occur if the connection is flipped. I have shown such a cable in the breadboard wiring view in **Figure 5**. Regardless of whether you use a cable type that re-arranges the wiring order, *be absolutely sure to observe correct polarity*. Or poof goes your sensor.

Using the Devantech SRF05 Ultrasonic Ranger

Ultrasonic distance measurement —also called ultrasonic ranging — is now an old science. Polaroid used it for years as an automatic focusing aid on their instant cameras. To measure distance, a short burst of ultrasonic sound (usually at a frequency of around 40 kHz) is sent out through a transducer; in this case, the transducer is a specially built ultrasonic speaker. The sound bounces off an object and the echo is received by the same or another transducer (this one is an ultrasonic microphone). A circuit then computes the time it took between the transmit pulse and the echo,

Listing 2 – usonic.pde.

```
int duration;                      // Stores duration of pulse
int distance;                      // Stores distance
int srfPin = 7;                    // SRF05 connected to dig pin D7

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(sonarRead(), DEC);  // Show in Serial Monitor
  delay(200);
}

int sonarRead () {
  pinMode(srfPin, OUTPUT);         // Set pin to OUTPUT
  digitalWrite(srfPin, LOW);       // Ensure pin is low
  delayMicroseconds(2);
  digitalWrite(srfPin, HIGH);      // Start ranging
  delayMicroseconds(10);           //  with 10 microsecond burst
  digitalWrite(srfPin, LOW);       // End ranging
  pinMode(srfPin, INPUT);          // Set pin to INPUT
  duration = pulseIn(srfPin, HIGH); // Read echo pulse
  distance = duration / 74 / 2;    // Convert to inches
  return(distance);                // Return value
}
```

and comes up with distance.

At sea level, sound travels at a speed of about 1,130 feet per second (about 344 meters per second) or 13,560 inches per second. This time varies depending on atmospheric conditions, including air pressure (which varies by altitude), temperature, and humidity.

The time it takes for the echo to be received is in microseconds if the object is within a few inches to a few feet of the robot. The overall time between transmit pulse and echo is divided by two to compensate for the round-trip travel time between robot and object.

Given a travel time of 13,560 inches per second for sound, it takes 73.7 µs (microseconds or 0.0000737 seconds) for sound to travel one inch. If an object is 10 inches away from the ultrasonic sensor, it takes 737 µs to travel there, plus an additional 737 µs to travel back, for a total "ping" time of 1,474 µs. The calculation is:

$$(1{,}474 / 73.7) / 2 = 10$$

First, divide the total transit time by 73.7 (use 74 to avoid using floating point math), then divide by 2. The result is the distance from sensor to object, in inches.

For the ArdBot, I've selected the SRF05. This sensor is among a series of ultrasonic rangers by UK-based Devantech, and were among the first to be marketed directly to robot experimenters. The SRF05 is relatively low-cost and easy to use, requiring just one signal line to the Arduino, in addition to power and ground.

There are, of course, other ultrasonic rangers that do pretty much the same thing. You can substitute if you'd like. There's the Parallax Ping (which is modeled after the SRF0x form factor), and various models by Maxbotix. All are excellent products and are well supported by their manufacturers.

Figure 6 is a schematic diagram for connecting the SRF05 to the Arduino, and **Figure 7** shows the same circuit in breadboard view. Hookup is simple — just route wires between the sensor and microcontroller. The SRF05 comes to you with plated-through holes for the electrical connections. You can solder wires directly to the board, or better yet use a five-pin male header. You can solder the header from either side. I like attaching the wires from the rear (opposite the transducers), so that they don't get in the way of the business-end of the sensor.

Be careful not to cross up the power and ground wires. Review the datasheet for the correct pinout. Note that there are two sets of five plated-through holes on the SRF05 circuit board. Don't use the set closest to the LED that's mounted on the back (component) side of the board. Those are used for programming the thing at the factory.

As used in the ArdBot, the SRF05 is set up in single-pin mode, using one pin (the one in the middle) for both Trigger and Echo. Refer to **Listing 2** for a demonstration sketch that shows how to take a reading from the SRF05 and display it in the serial monitor window. The value returned by the *readSonar()* function is converted to whole inches. According to its documentation, the SRF05 can measure distances from 1 cm to four meters. That equates to a range of 0.4 inches to 157 inches.

Adding a Sensor Turret

Anchoring a sensor directly to a robot has its disadvantages. The bot can only "see" what's directly in front of it. To widen its field of view and see at other angles, the robot must literally reorient itself — something akin to spinning around while walking. Okay for the Ministry of Silly Walks, but immensely silly for robots.

With a motorized sensor turret, your robot can scan its environment without ever taking a step — or rolling a wheel. One or more sensors are attached to an R/C servo motor which is commanded to sweep back and forth while the sensors send back data.

Accuracy is somewhat diminished when using an ultrasonic or infrared sensor when it's attached to a rotating sensor turret. Movement affects the reading. During any motion, you can use the measurements for general proximity detection, but you may wish to momentarily stop the turret (and robot) to get a more accurate distance reading.

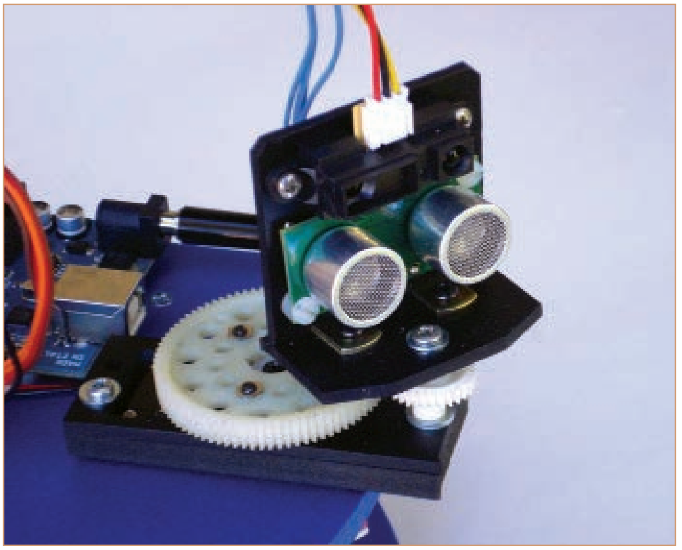


FIGURE 8. A servo turret capable of a full 360 degree rotation. (See the text for details.) You can use a standard servo turret (capable of 180 degree rotation) if you prefer.

Figure 8 shows a sensor turret with both Sharp IR and ultrasonic modules attached to it. This particular turret (from Budget Robotics; see the **Sources** box) can spin the sensors a full 360 degrees, enabling the bot to literally see what's in front, to the sides, and behind. You can use a standard servo-driven turret which provides up to 180 degrees of side-to-side vision.

You'll want to select a sensor mount that accepts both the IR and ultrasonic detectors. The mount in the picture is made of lightweight plastic, keeping down the overall weight of the robot. Plastic also doesn't contribute to microphonic ringing that can be caused whenever the ultrasound module does its "ping" thing. When using a metal mount, you may wish to add small rubber grommets to the mounting screws — they're included with most R/C servos — as a form of insulator. Be sure the exposed back of the sensor does not make direct contact with the metal, or else a short circuit could occur.

Listing 3 shows a demo sketch for rotating the turret in small segments of an arc, sweeping from one side to the other. I've intentionally limited the span of the turret to prevent the wires from getting tangled, and from having either sensor deliver inaccurate results because of the wire harness at the rear. As shown in the sketch, the turret has an approximate 250 degree span. If you use a 360 degree turret, you can experiment with larger or smaller arcs.

Refer to **Figures 9** and **10** for a schematic and breadboard view, respectively, of connecting the turret servo to the Arduino. Note that the breadboard view assumes the lower half of the breadboard is already populated with the wiring described in Part 2 of this series.

As a demonstration, the turret is made to stop periodically every 25 degrees, or 10 times when going

Listing 3 – turretScan.pde.

```
#include <Servo.h>

Servo myServo;
const int delayTime = 2000;
const int servoPin = 8;

void setup() {
  myServo.attach(servoPin);
}

void loop() {
  turretL();
  turretR();
  // Rotate turret in steps
  //  Arguments: min-position, max-position,
  //  delay between steps
  turretStep(40, 140, 150);
  turretCenter();
  delay(delayTime);
}

void turretR() {
  myServo.write(40);
  delay(delayTime);
}

void turretL() {
  myServo.write(140);
  delay(delayTime);
}

void turretStep(int minVal, int maxVal, int
stepDelay) {
  for(int i=minVal; i<=maxVal; i+=10) {
    myServo.write(i);
    delay(stepDelay);
  }
}

void turretCenter() {
  myServo.write(90);
  delay(delayTime);
}
```

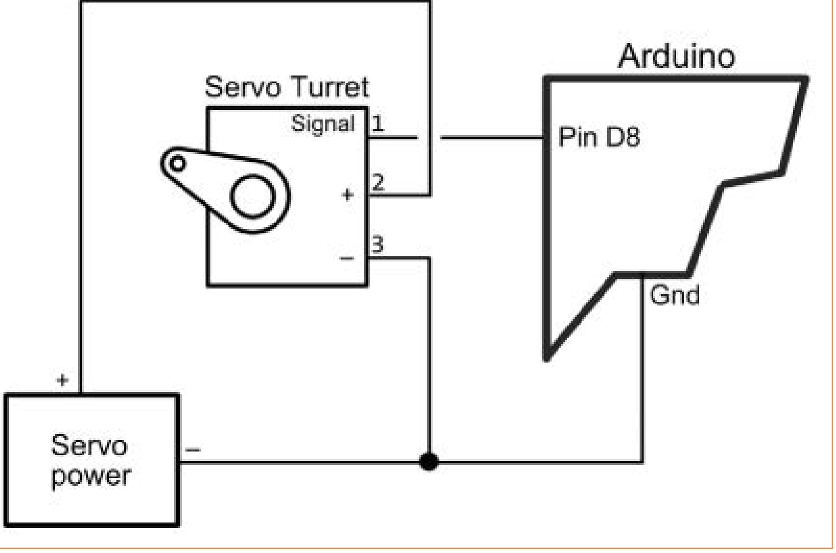


FIGURE 9. Connection schematic for the servo turret. Note the use of the separate power supply for the servo.

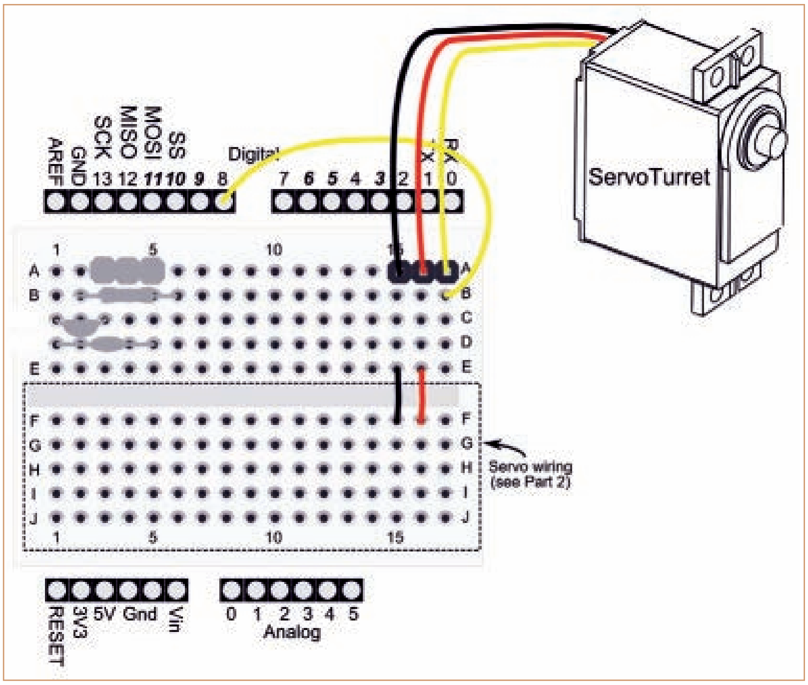


FIGURE 10. Breadboard view of connecting the servo turret to the Arduino. Note that you need to already have the servo wiring in place (from Part 2 of this series) in the bottom half of the breadboard.

from one extreme to the other. During each stop, you'd take a reading from either or both of the IR and ultrasonic sensors. Next month, when there's a little more page space to play with, we'll examine an all-in-one "wanderbot" sketch that combines the sensor turret motion, infrared proximity readings, and ultrasonic pings.

Notice that for the 360 degree turret and a GWS S03 servo, the *turretR()* and *turretL()* functions work in reverse. The *turretR()* function actually scans to the left, and *turretL()* scans to the right. This is because the gearing used on the turret flips the direction of travel of the servo. I left it this way in case you use a regular non-360 enhanced turret. Otherwise, you can simply rename the functions to flip their logic.

Coming Up

Next month, we'll add a line following module, and conclude with putting what we've learned to good use. You'll find ideas and programming code for your ArdBot to wander about a room, looking for people, pets, and things to investigate. **SV**

Gordon McComb can be reached at arduino@robotoid.com.

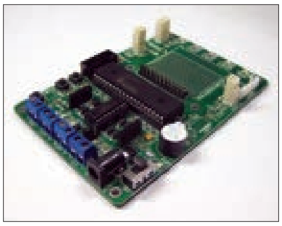
Continued from page 21

many different platforms.

FlashFly will begin shipping in March '11.

Blue Wolf Website: **www.bluewolfinc.com**

Low Cost Platform



Cytron Technologies is now offering the MC40A — a low cost platform for beginners to start developing autonomous robots. It is ready to use with all the basic features onboard, such as a PIC16F887; motor driver L293D which supports two brush motors; two programmable pushbuttons; a buzzer; two digital inputs; one ADC input; LSS05 (line sensor bar) connector, plus more. The MC40A is designed as a mini mobile robot controller. It comes with sample source code for the PIC16F; a user may start immediately with powering up the controller. Some of the features include:

- Suitable for 40-pin eight-bit PICs including the PIC16F and PIC18F.
- Comes ready with PIC16F887.
- Sample source code for line following, ADC, LCD, DC brush motor, and UART.
- Input power: 7V to 12V.
- Supports two DC brush motors (max 1A per motor).
- Motor power: 5V-12V; selectable from Vmotor or share input power.
- Supports 2x8 parallel LCD (optional).
- Supports Cytron SK including SKPS, SKXBee, and SKKCA.
- Ready with two connectors for limit switches.
- Ready with ADC input for an infrared distance sensor or ultrasonic distance sensor.
- Ready with LSS05 (auto-calibrating line sensor) connector.
- Ready with ICSP connector for UIC00A/B for loading program to PIC.
- Ready with connector for UC00A, USB-to-UART converter.
- Free I/O pins are extended out for further development.

For further information, please contact:

Cytron Technologies Website: **www.cytron.com.my**

NEW LAUNCH

Launch of New Website

After five years of development, MINDS-i is offering their unique patented quick-lock construction system for purchase over the Internet. Inventor Mike Marzetta is excited to unveil the culmination of his vision at **www.mymindsi.com**.

"We are excited to enter the Internet market with **www.mymindsi.com**. The site is the launch point for consumers to purchase our RC, robotic, and alternate energy construction kits. The website allows us to make MINDS-i available to customers nationwide" said Marzetta. Previously, MINDS-i was only available by special order. The website features basic construction kits, advanced kits (with and without electronics), and accessories. The wide variety of online products further enables anyone to build, create, and then re-create anything they can envision in their "mind's eye," with simple manipulation.

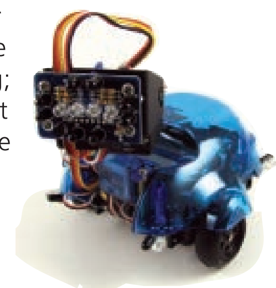
Larry Bernstein, MINDS-i CEO and former President of Hasbro's Toy Division, "sees the potential of MINDS-i as a true paradigm shift. MINDS-i is unveiling something that simply does not exist in today's marketplace."

For further information, please contact:

MINDS-i, Inc.
22819 E Appleway Ave.
Liberty Lake, WA 99019
Tel: **509 • 252 • 5767**
Email: **info@mymindsi.com**
Website: **www.mymindsi.com**

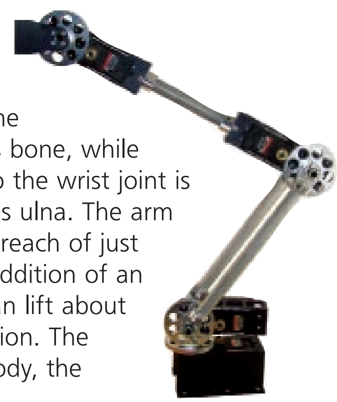
ROBOT KITS

Dagu Adventure Robot Kit



The Dagu Adventure Robot Kit available from RobotShop is a small robot designed to wander around looking for moving objects. This kit is ideal for beginners and students. Adventure Bot requires no soldering or wiring; it comes pre-programmed. You just need to add 4x Ni-Mh rechargeable batteries. Once the batteries have been installed, the robot can be recharged using a standard 9V power pack. Four corner sensors help the robot avoid collisions. The IR compound eye mounted on a small pan/tilt assembly allows the robot to track the movement of nearby objects and can help the robot judge distance.

Modular Robotic Arm Kit



RobotShop's M100RAK Modular Robotic Arm Kit is one of the first affordable and versatile "intermediate sized" robotic arms available. It is loosely modeled on the human arm; the length of the shoulder joint to the elbow joint is almost the same length as an adult's humerus bone, while the length from the elbow to the wrist joint is almost the same as an adult's ulna. The arm has a maximum (suggested) reach of just over 24 inches without the addition of an end effector (gripper) and can lift about 500 g (1.1 lbs) at this extension. The closer the weight is to the body, the more the arm can lift.

For further information on these items, please contact:

RobotShop Website: **www.robotshop.com**

CPLDs — Part 1

complex programmable logic devices

An Introduction

by David A. Ward

Anyone involved in the field of digital electronics should become familiar with CPLDs or complex programmable logic devices. This article and those that follow, will help you get started implementing CPLDs into your circuit designs. CPLDs are ICs that can be programmed to replace many standard 74XX series ICs, as well as perform any logic functions that the 74XX series can perform when combined together. The CPLD that will be explored in these articles is the Xilinx XC9572XL in a 44-pin PLCC (plastic leadless chip carrier) which can incorporate up to 1,600 logic gates in a single IC.

Although there are several manufacturers of CPLDs, these articles will only deal with the Xilinx brand. There are several reasons why Xilinx CPLDs are used. First, Xilinx provides their design, simulation, and programming software (Xilinx ISE) for free. Second, you can purchase a Xilinx CPLD programmer for $129; $89 for academic pricing. Third, you can purchase the XC9572XL CPLD in a 44-pin PLCC package which can readily be breadboarded using a standard 0.1" breadboard and PLCC adapter board. Finally, Xilinx is one of the largest — if not the largest — manufacturer of CPLDs in the world, so learning how to use their products is a good path to start down.

Before we delve into the XC9572XL CPLD, let's take a brief look at the history of PLDs or programmable logic devices. In about 1978, PALs were introduced. These PALs — or programmable array logic devices — were OTPs or one time programmable devices. Once you programmed or "burned" your logic into them, they could not be erased and reprogrammed. Next came GALs — or generic array logic devices — in about 1985. GALs are similar to PALs but can be reprogrammed. CPLDs are essentially several GALs inside of one IC, allowing for a much higher logic gate count. PALs and GALs might be able to incorporate several hundred logic gates whereas the Xilinx XC9572XL CPLD can incorporate up to 1,600 of them. This CPLD also uses "fast Flash technology" to retain the internal logic connections. This means that this CPLD is non-volatile; it won't lose its information or configuration when power is removed from the chip. It also means that it can be erased and programmed up to 10,000 times. Xilinx datasheets rate the XC9572XL CPLD data retention at 20 years.

Let's take a look at **Figure 1** to see what's inside this CPLD and see how it functions. The XC9572XL CPLD contains 72 macrocells (that's where the 72 in its part number comes from); four function blocks times 18 macrocells per function block. A macrocell consists of an "SOP" or sum of products structure. Each macrocell is capable of up to 54 inputs and 18 outputs. The SOP structure consists of arrays of AND gates with programmable interconnections. This forms the "products" part of the SOP structure as in **Figure 2**. Looking at the AND gate truth table in **Figure 3**, you can see that it acts or appears like multiplication. Several of the AND gates are then connected to an OR gate for the "sum" portion of the SOP structure. Looking at the OR gate truth table in **Figure 4**, you can see that it acts or appears like addition or summing of the 1s and 0s.

Perhaps studying a simplified diagram of a CPLD — without all of the details — will help clarify things a little.

Take a look at **Figure 5**. Notice that all of the inputs can come in as they are being inverted. Using the inverters, AND gates and OR gates (all digital logic gates) can be configured: AND, OR , NOT, NAND, NOR, XOR, and XNOR, as well as any logic functions that these gates combined together can create. When digital signals appear on the inputs of the CPLD, they go through the SOP structure and the resulting logic condition is sent out to the output pins within 5 nS (five billionths of a second).

Anytime the input logic changes, the output will respond within 5 nS; you can purchase slower CPLDs (7.5 nS and 10 nS) for a little bit less. CPLDs do not operate like microprocessors; they do not need a clock signal to operate, although clock signals can be used to control the logic when desired. CPLDs operate in a truly "parallel" or concurrent manner, whereas microprocessors operate in a sequential or step-by-step manner. This makes CPLDs extremely fast when compared to 74XX series ICs and microprocessors.

There are several reasons why CPLDs are preferred over 74XX series chips and microprocessors for certain functions. First is cost. If the XC9572XL CPLD costs about $2 for 1,600 logic gates, how would that compare to 74XX series ICs? If a typical 74XX series chip costs $0.50 and contains four logic gates, 1,600/4 = 400 chips; 400 X $0.50 = $200. Secondly, consider real estate savings on printed circuit boards (PCBs). How much physical space would 400 DIP or even SOIC surface-mount 74XX ICs take compared to one 44-pin PLCC? If the 44-pin PLCC took up as much space as four DIP chips, that would mean a space savings of about 100 times.

Third is reliability. How many external physical connections would be saved by using the CPLD over the 400 74XX chips? Perhaps thousands, certainly hundreds. Fourth is flexibility. If the circuit needed any changes, the CPLD can simply be reprogrammed, even while it is still in the circuit. Any changes to the 74XX circuit with 400 chips could require a complete redesign of the PCB. Fifth is speed. The logic operations within the CPLD will still take 5 nS, whereas the 74XX circuit's propagation delays will add up as signals pass from one 74XX gate to the next. Finally is power savings. One CPLD's power consumption versus 400 74XX ICs would be much, much less.

If you want to take a closer look at the XC9572XL CPLD, you can download the datasheets from **www.xilinx.com** and do a search for DS057. However, we

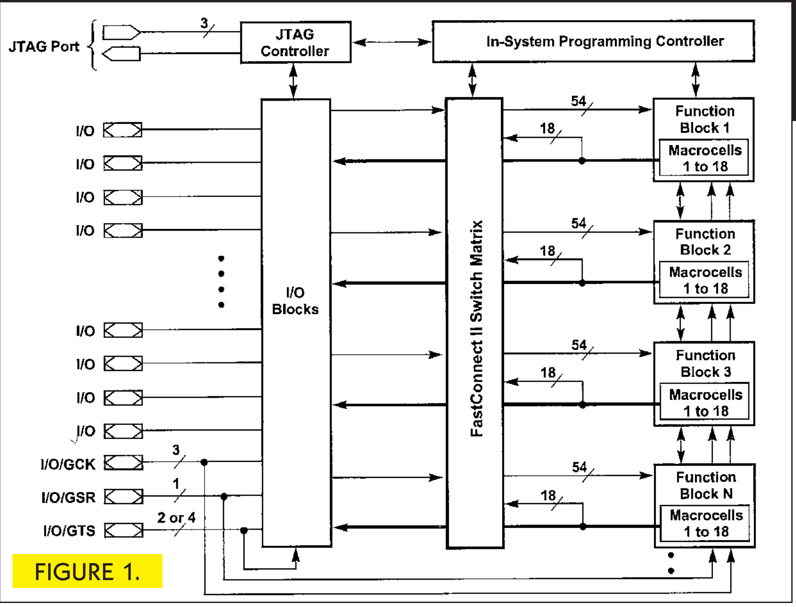


FIGURE 1.

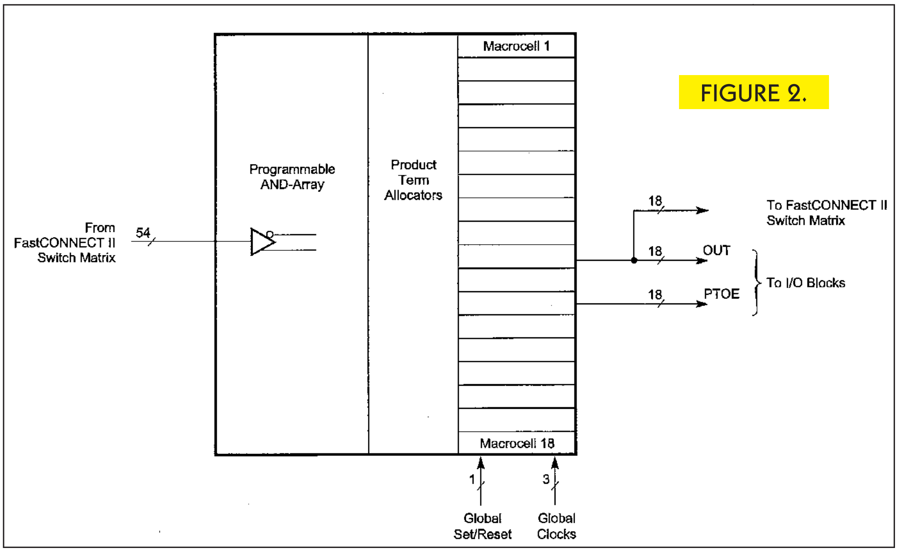


FIGURE 2.

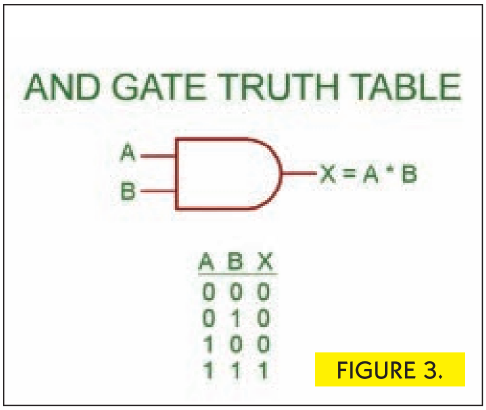


FIGURE 3.

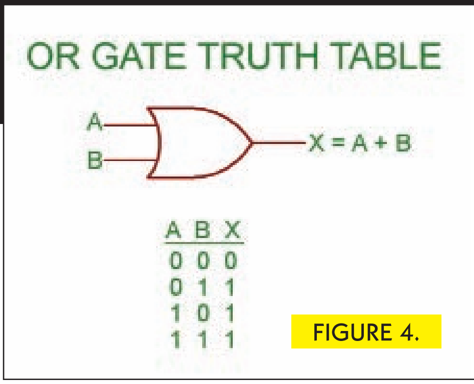


FIGURE 4.

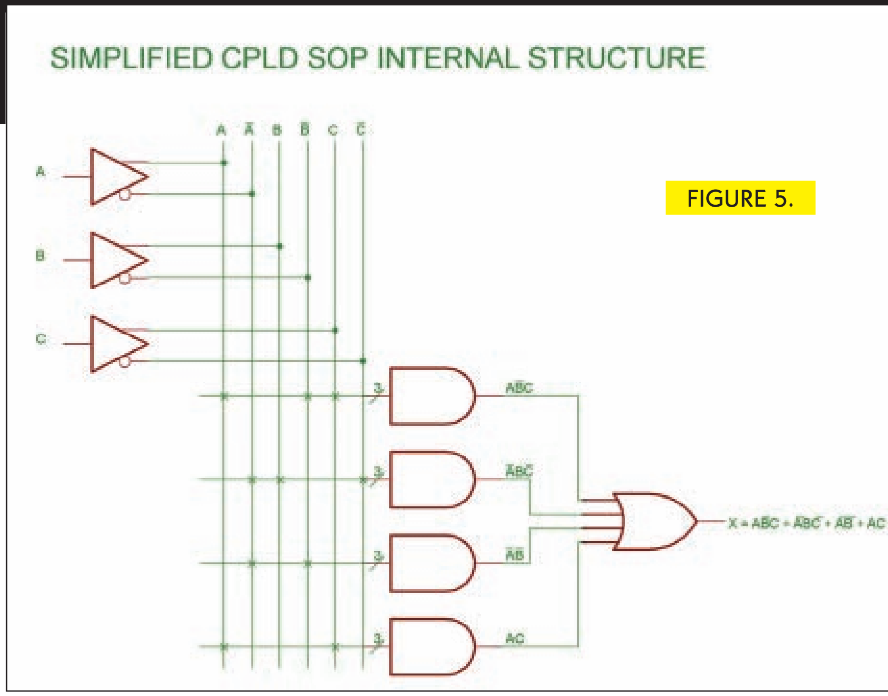


FIGURE 5.

will cover a few of the major points at this time.

The XC95872XL CPLD in a 44-pin PLCC package has 34 user I/O (input/output) pins. The remaining 10 pins are not available as user I/Os since they are used for power and programming. Note that the drawing of a CPLD package in the datasheets is not a 44-pin PLCC package but a 144-pin surface-mount package. **Figure 6** is a drawing of a 44-pin PLCC package. Pin 1 is located above the "dot" on the package.

The pin numbering proceeds in a counter-clockwise manner. The pin to the left of pin 1 is pin 2; the pin to the right of pin 1 is pin 44. Note also that there is a slightly larger chamfered corner on the upper left-hand corner of the package to help when positioning the chip in its socket. Although with a little extra misguided effort, you can force the chip into the socket in any direction — a very bad idea.

Pins 21 and 41 connect to +3.3V and are labeled VCCINT, VCC internal. Pins 10, 23, and 31 connect to ground and are labeled GND. Pin 32 is labeled VCCIO, VCC input/output, and can be connected to either +2.5V or +3.3V, depending on what you want the voltage to be when an output pin is high or a '1.' Four pins are used to program and communicate with the CPLD through the JTAG protocol to the programmer. Pin 17 is labeled TCK or test clock. Pin 15 is labeled TDI or test data in. Pin 30 is labeled TDO or test data out. Pin 16 is labeled TMS or test mode select. The programmer has another connection labeled Vref — or reference voltage — which connects to +3.3V and a ground wire, as well.

Perhaps you've heard about FPGAs or field programmable gate arrays. Let's take a minute to compare CPLDs and FPGAs. FPGAs are the latest development in the PLD field; they perform all of the same functions that a

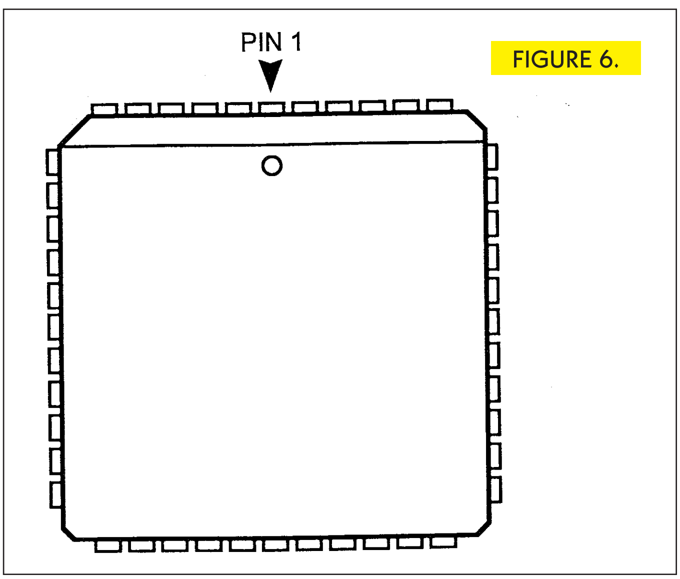


FIGURE 6.

TABLE 1. CPLD Part list

	Description	Source	Part #	Cost
1	Xilinx XC9572XL CPLD, 44 PLCC, 5 nS	www.digikey.com	122-1466-ND	$2.04
2	3.3V Voltage Regulator, TO-220	www.digikey.com	LM2937-3.3-ND	$1.98
3	Xilinx USB-JTAG Programming Cable	www.digilentinc.com	XUB-USB-JTAG	$129.00/ $89.00 Academic
4	44 PLCC to Proto-board Adapter	www.jameco.com	73542	$13.95
5	PLCC Extractor	www.jameco.com	16766	$3.95

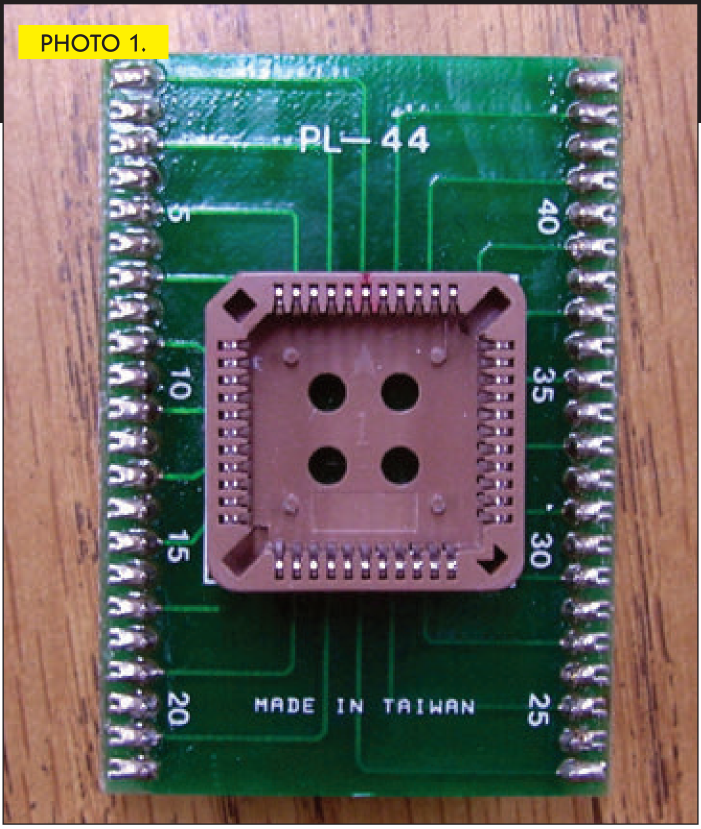
www.servomagazine.com/index.php?/magazine/article/march2011_Ward

CPLD does, as well as a few more. FPGAs are built differently inside than CPLDs are. Most importantly, they are volatile; that is, their logic configuration must be stored in an external non-volatile memory device and loaded into the FPGA on every power-up cycle. FPGAs can be programmed with the same Xilinx software through either the graphical schematic method or using HDL — hardware description language.

So, why start with CPLDs instead of FPGAs? One big reason is that most FPGAs come in surface-mount packages so you really need to purchase an FPGA development board. Next, you'll have to deal with an external memory device which adds complexity to the situation. FPGAs are quite a bit more expensive than CPLDs: $2 versus $15. However, everything you learn about CPLDs will transfer to FPGAs, and since you can readily breadboard the CPLD and work with a single chip, it seems like a good place to start.

Next month, you'll learn how to program the CPLD. The Xilinx ISE software will allow you to program your CPLDs in two major ways: graphically through a schematic diagram or through HDL. The next article will illustrate entering a logic program through the graphical method and then programming a CPLD. The third article will illustrate setting up and running a simulation. The fourth article will illustrate using HDL to enter a logic design. The fifth and final article will demonstrate incorporating a CPLD to control a mobile robotics platform.

PHOTO 1.



Before you can begin programming your CPLD, you'll need to purchase several items; see **Table 1** and **Photos 1**, **2**, and **3**. While you're waiting for your parts, you can go to **www.xilinx.com** and download and install their free webpack version of their latest ISE software. SV

PHOTO 2.

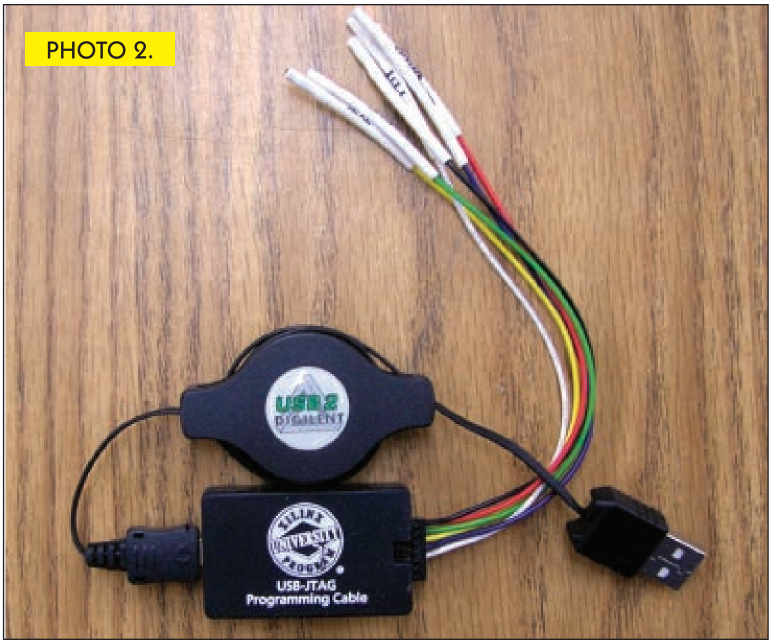


PHOTO 3.

Build Your Own Big Walker Part 2

Mechanical Design

by Daniel Albert

www.servomagazine.com/index.php?/magazine/article/march2011_Albert

To recap last month's article, this big walker uses an inverse pendulum weight distribution as opposed to big heavy feet. An inverted pendulum has its mass above its pivot point, so it is inherently unstable. The goal is to use the three weight sensors (load cells) on each foot to feed back weight information to dynamically balance the mass. Each leg has a micro control unit (MCU) that controls the five servos of each limb to always keep the weight at the centroid. This month, I will discuss the design details of the biped's feet, ankles, and hips.

Balancing on Three Points

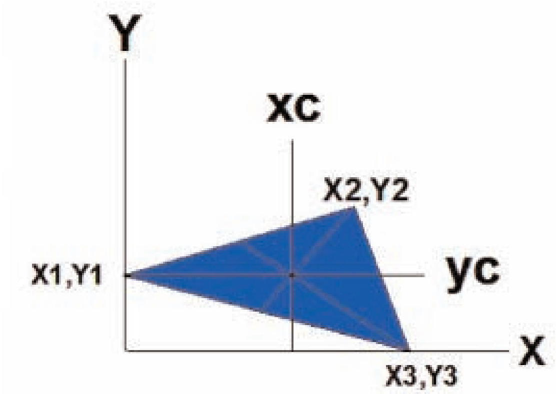
A tripod is an extremely stable structure. On a level surface with three legs of identical length, a tripod will center the weight of a mass that is set onto the point where the legs intersect. Picture a camera on a tripod. If the surface tilts up, down, or sideways, the weight will move from the center.

By adjusting the length of one or two legs, we can re-center the mass, thus making the structure stable. With the load cell measurements at each of the three points, we can calculate the direction to adjust each leg length to re-center the weight. We could also calculate the amount to adjust the length. This design does not require the scale of the error. Each adjustment is the minimum movement. Calculations and adjustments happen every 50 mSecs (20 times a second). The system will dynamically stay in balance as long as the angles of the surface do not change too fast.

FIGURE 1. Centroid triangle.

$$XC = \frac{(M1 * X1) + (M2 * X2) + (M3 * X3)}{M1 + M2 + M3}$$

$$YC = \frac{(M1 * Y1) + (M2 * Y2) + (M3 * Y3)}{M1 + M2 + M3}$$



The three tripod points on each foot distribute and measure the weight of the leg individually. These correspond to the heel, the ball of the first metatarsal at the big toe, and the fifth metatarsal at the pinkie toe.

We use only two lengths of the tripod to tilt and roll the foot, so we only need two servos.

To keep the weight closer to the hip, the foot

PHOTO 1. Foot bottom.

is controlled by linkage instead of the traditionally mounted ankle servos. The linkage connects the bottom of the foot to two servos mounted at the knee via universal joints. The third leg of the tripod is connected to only one universal joint at the heal.

Calculating the Center of Gravity:
(CG) = Center of Mass (CM)
In a uniform gravitational field, these terms are the same.

Our three points are in two-dimensional space, so we only need the weight, X, and Y data to solve. The points never change their distance from each other. This makes x1, x2, x3, y1, y2, and y3 constants. Only the weights (m1, m2, and m3) are variable (see **Figure 1**). Solving for XC and YC gives us the center of the mass (centroid) for three mass points.

I have rotated the triangle to match the load cells on the foot (see **Photo 1**). X1,Y1 corresponds to the heel; X3,Y3 to the first metatarsal; and X2,Y2 to the fifth metatarsal.

An additional feature of this orientation is that it easily allows the linkage to directly match the CG error. Any weight that is not at YC and XC is an error. Y errors above or below YC require a left or right sideways lean correction. X errors above or below XC require a forward or backward lean correction.

The linkage (**Photo 2**) translates adjustments from the two servos to control the foot motion. Universal joints allow for three-dimensional movement of each linkage. The heal linkage does not change length but merely rotates. When both metatarsal linkages push down, the robot leans back and the weight shifts along the X axis towards zero. When both linkages

PHOTO 2. Linkage.

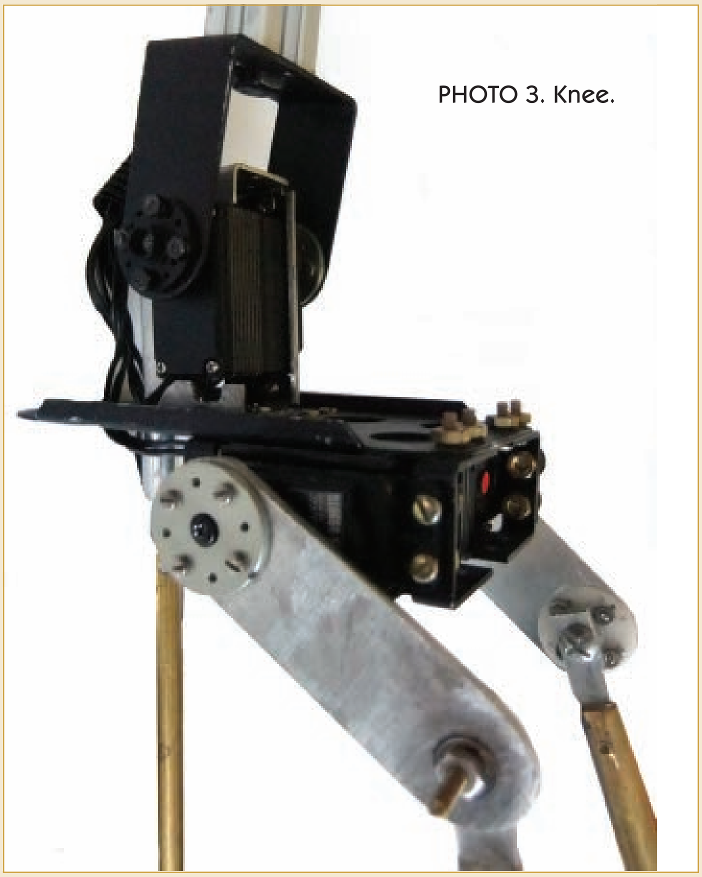
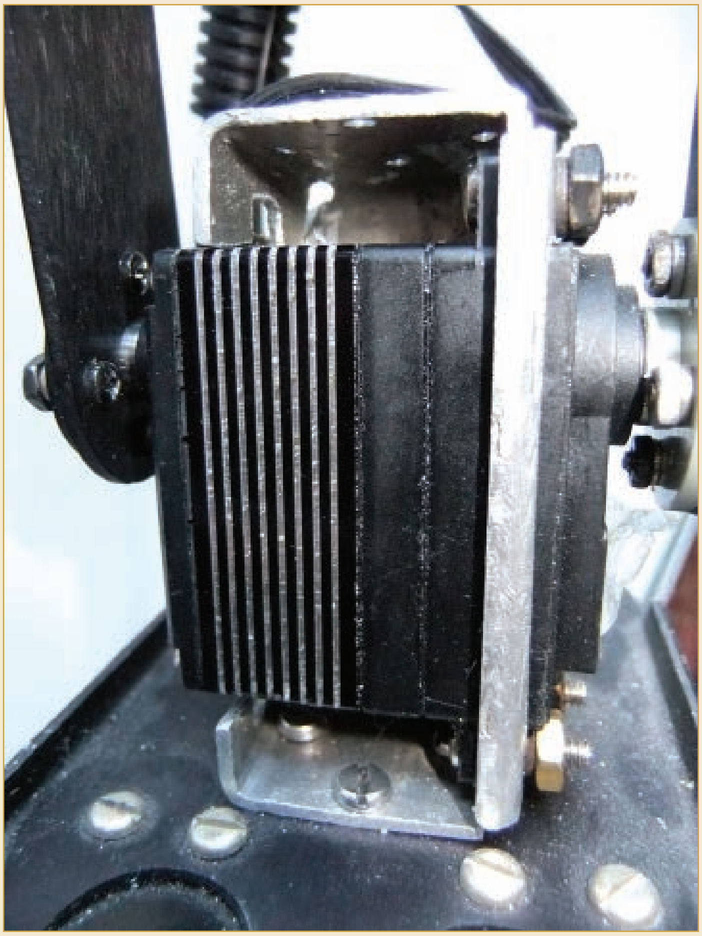


PHOTO 3. Knee.



pull up, then the robot will lean forward as the weight shifts along the X axis until a face plant occurs. Opposite linkage movements of one up and one down will move the weight along the Y axis.

Servos

I needed some very strong servos for this project. I also wanted to keep the cost low. The Hitec HSR5990TG fits both requirements. They can run up to 7.4 volts and have a stall torque of 30 kg/cm or 416 oz/in. At about $100 each in quantity, they also fit my budget.

Rather than build every part from scratch, I found it very useful to buy pre-made servo mounting brackets. Lynxmotion has a good selection of mounts, brackets, and hardware. When working on a project like this, I found it saved a great deal of time to just buy two of everything that was under $10 each. Parts that I thought would work didn't, and parts I just bought because they looked cool ended up being perfect.

For example, the base of the knee was built from a biped foot plate. Having it in my stock of parts saved me hours of fabricating this part (see **Photo 3**).

Hobby shops and even Ace Hardware sell brass tubing that comes in 12 inch lengths and different inner diameters. Once again, buy two of each and keep them in your stock. You don't want to have to stop what you are doing to run out to the store to buy a $2 part. Some simple parts you just can't find for sale, so it's good to have lots of scrap aluminum pieces. The connecting piece from the servo horn to the foot linkage took two hours to make. Of course, several months after I made the first two I found that LEGO had the perfect piece.

While these servos are the standard R/C form factor, that does not bode well with biped robot design. They do come with a replaceable rear plate that allows a bracket with a bearing to be mounted on the opposite side of the horn, however. This allows a Lynxmotion "C" servo bracket to connect and form a solid one degree-of-freedom joint (see **Photo 4**).

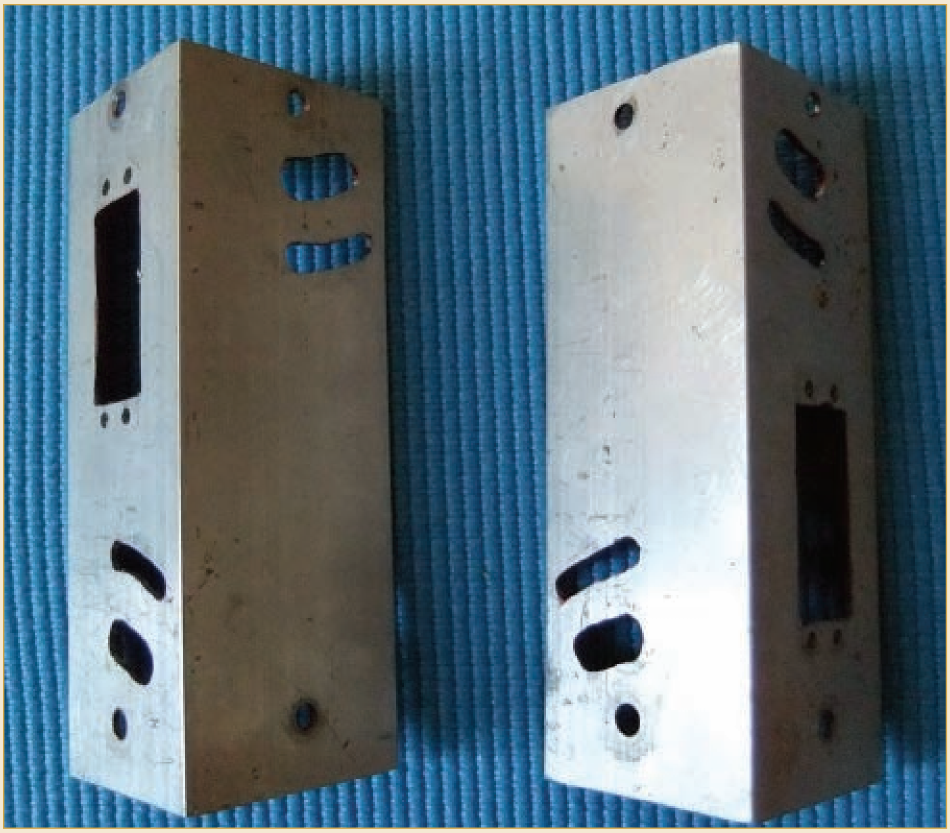
At the time, there was not a ready-built bracket to mount the servo to the knee base, so I fabricated a two piece bracket that held the knee servo. Most brackets have one or two 90 degree bends. They are stamped from flat metal and bent to allow mounting in planes other than the one where the servo mounts are located. They seem strong, but after a long beam is attached there is very noticeable movement in the bends unless a solid box is formed. Two C shaped brackets bolted together at the open ends creates a solid box.

Belt Driven Gear Box

The full weight of the leg off the ground is supported by the opposite hip. The torque at the hip was too much

PHOTO 4. Knee servo bracket.

PHOTO 5. Transmission bracket.



for these servos. Another very critical problem was the fragility of the gears in the servo. If the robot took a tumble, it was very likely that the servo gears would be damaged. Both problems were solved with a belt driven transmission. In this case, I was willing to trade speed for torque. So, I decided to gear the drive down 3-to-1 for more torque and therefore less speed. This was not a problem since the hip servos never move more than about 20 degrees. The spec for the HSR5990TG is .14 secs for 60 degrees with no load. Under heavy load, it is slower. With the transmission, it is even slower. It can still travel the full 20 degrees in under a second, however.

Building the transmission was an exercise in custom fabrication. The parts were all available online, but not from one place. Once again, Lynxmotion came through with an MXL timing pulley that attached to the servo, but I needed to hunt down the belts and other pulleys. I used a standard 1/4 inch shaft, bearing, and pulleys. This gave great strength to the joint. The belt helps isolate the servo from the shaft pulley for when the bot falls. I have a small Sherline mill that was perfect for making the mounting brackets. Several years back, I added CNC capability.

With a little G-code (used for instructing the mill), I got near perfect square openings and perfectly lined up holes for mounting the servos. Sometimes —

PHOTO 7. Batteries.

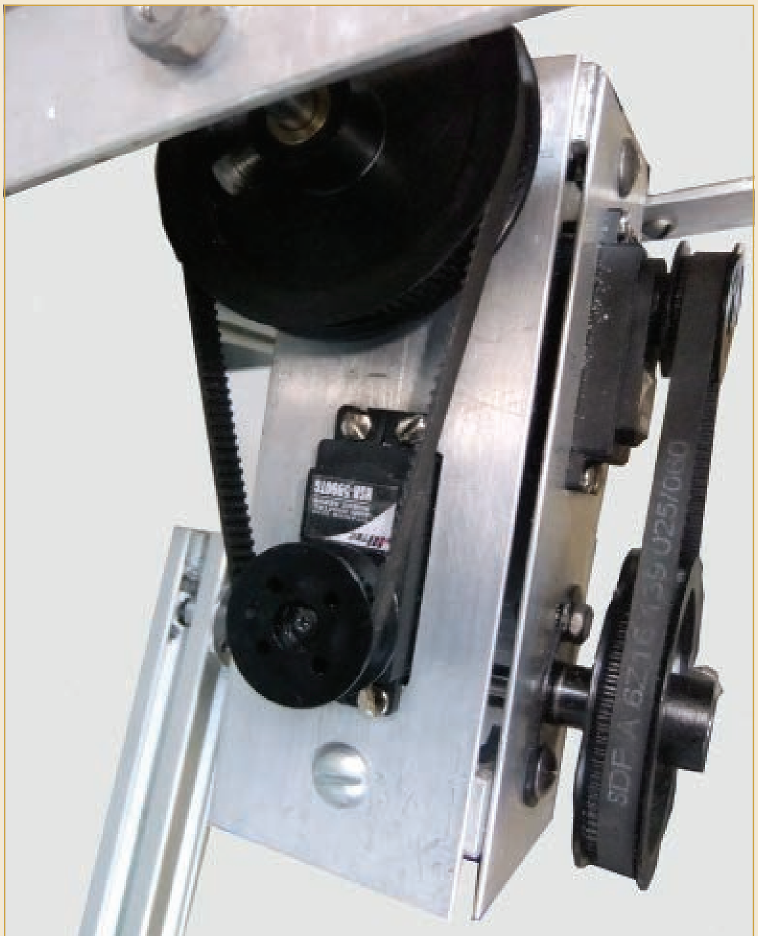


PHOTO 6. Hip gear box.

especially for one-off parts — milling is easier by hand. Of course, you don't get those nice cuts as can be seen by my not so perfect belt adjustment slots (see **Photo 5**).

The belts need to be tight to prevent the timing belt notches from slipping during operation, but slippage that prevents damage is okay if the bot falls. I would have preferred a larger pitch than the MXL pitch, but that was the only pulley with a Hitec spline mating.

The hip transmissions are mounted and rotate on the hip frame via the 1/4 inch shaft and bearings. Each transmission drives a leg forward and backward for a total of two degrees of freedom at the hip for each leg (see **Photo 6**).

For connecting the knee servo to the hip transmission, I used some one inch aluminum structural framing material. There are a number of companies that sell this including McMaster-Carr. This stuff is great for robotics. As an extruded material, much of the interior of the metal is hollow. This allows for light weight and extreme stiffness. It is also easy to drill and tap. I used a couple of 1/4-20 short bolts to lock it tightly to the knee.

At the hip side, I took a Lynxmotion aluminum tub connector hub and honed it out for the 1/4 inch shaft. I drilled and tapped a locking screw to hold it firmly to a flat I milled into the shaft. The hub easily mounted onto the framing material with four 2-56 1.5 inch screws and nuts.

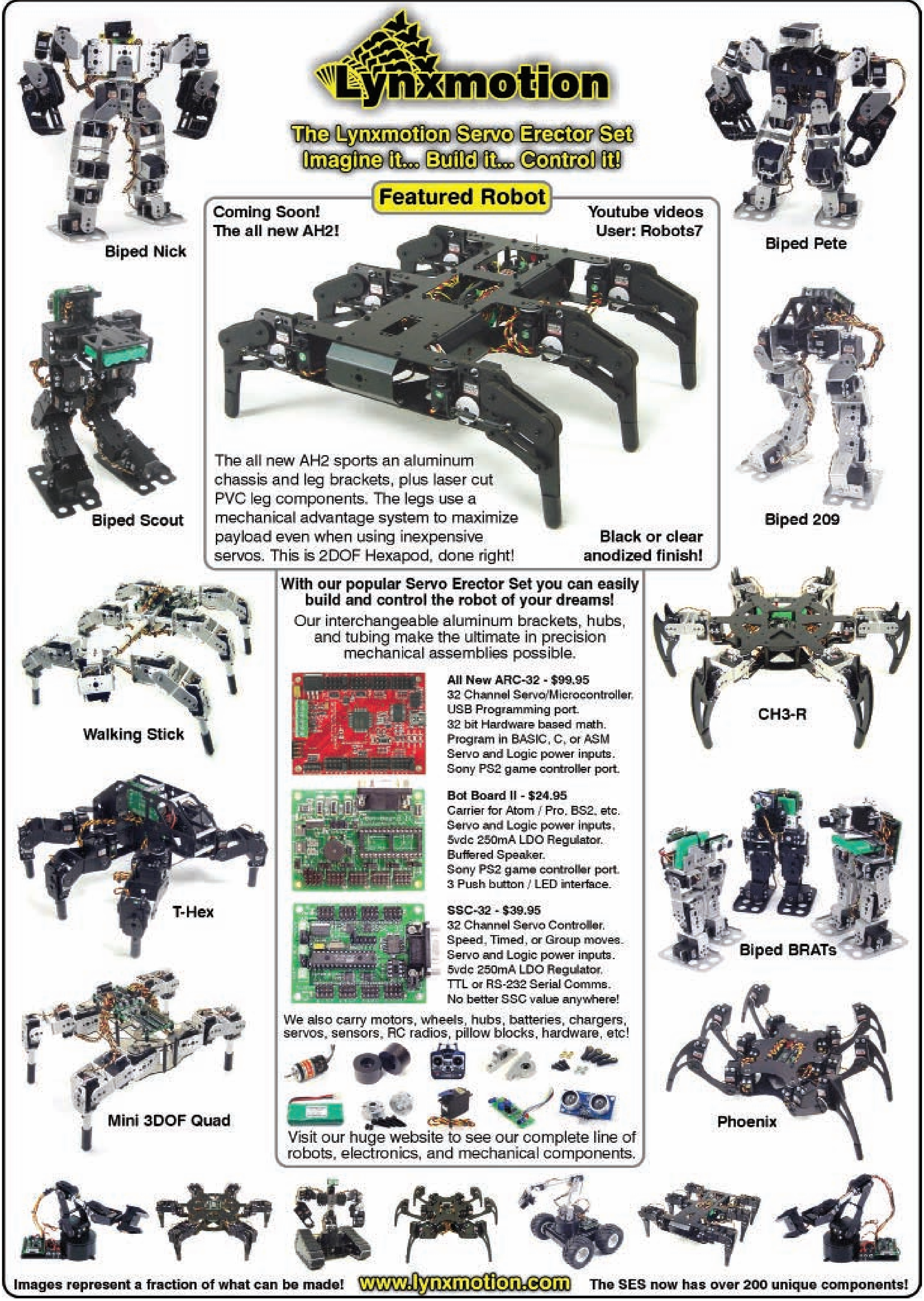
Power System

These are some powerful servos. The stall current is 5,200 mAmps. While it is unlikely that all 10 servos will stall, one or two will from time to time approach the stall current. Much of this is determined by balance in the system, but it never hurts to have enough power to arc-weld.

These particular polymer Li-Ion batteries from Tenergy have a discharge rate of 10 amps (1C). I put two in series and two in parallel for 20 amps at 7.4 volts (see **Photo 7**). In practice, this runs the bot for at least 30 minutes. They are fuse protected just before the main power switch. Wires can break and short, so the closer the fuse to the battery, the better.

Next Month

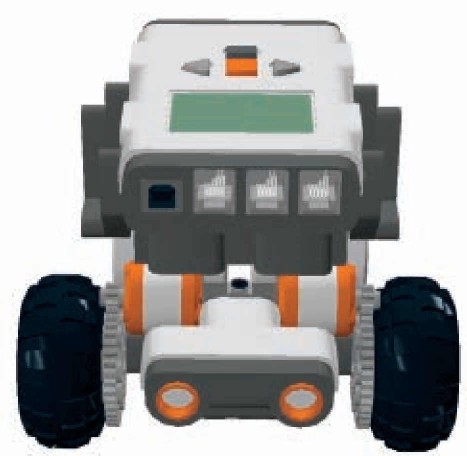
In the next segment, I'll describe the design and implementation of the distributed processing, multi processor system used to control each limb and the distributed communications that are necessary to have the limbs work in concert to walk. **SV**


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
Coming Soon!
The all new AH2!
Youtube videos
User: Robots7
The all new AH2 sports an aluminum chassis and leg brackets, plus laser cut PVC leg components. The legs use a mechanical advantage system to maximize payload even when using inexpensive servos. This is 2DOF Hexapod, done right!
Black or clear anodized finish!
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
All New ARC-32 - $99.95
32 Channel Servo/Microcontroller.
USB Programming port.
32 bit Hardware based math.
Program in BASIC, C, or ASM
Servo and Logic power inputs.
Sony PS2 game controller port.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Walking Stick
CH3-R
T-Hex
Biped BRATs
Mini 3DOF Quad
Phoenix
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!

The NXT Big Thing #8

Stay on Course!

By Greg Intermaggio



In the last edition of The NXT Big Thing, we made a major improvement to the standard Sumo robotics program by way of adding an ultrasonic sensor to Eddie 2.0 to detect the opponent robot. This time, we're changing gears, or rather, sensors. We'll be learning about the compass sensor available from **HiTechnic.com**.

www.servomagazine.com/index.php?/magazine/article/march2011_Intermaggio

This article assumes that you've already purchased a compass sensor from HiTechnic, and downloaded and imported the compass sensor block available from their website. Please contact HiTechnic with any questions regarding compass sensor troubleshooting.

Also, in the past we've plugged our left wheel into motor port B and the right into motor port C. This time, we'll be reversing them. Plug your left motor into port C and your right into port B before you forget.

All that said, let's get snappin'!

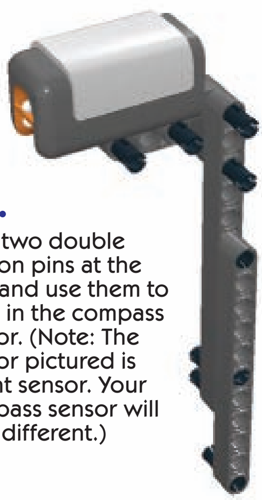
Building Instructions: Compass Sensor Attachment

1. Start with a 13-hole studless beam.

2. Attach a 3 x 5 studless beam to the top.

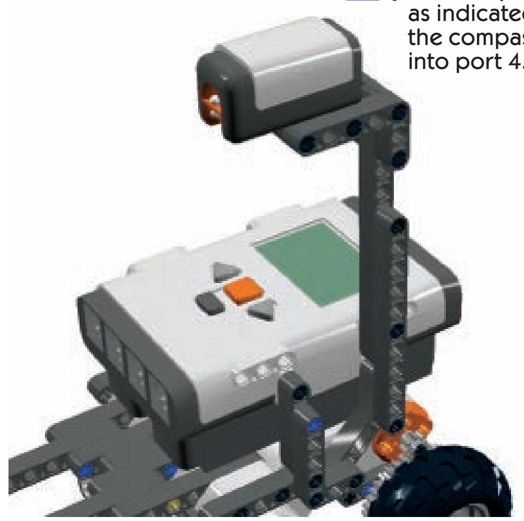
3. Snap an 11-hole studless beam as indicated.

4. Add two double friction pins at the top, and use them to snap in the compass sensor. (Note: The sensor pictured is a light sensor. Your compass sensor will look different.)



5. Snap on a 3 x 5 studless beam.

6. Attach the entire assembly to Eddie as indicated. Plug the compass sensor into port 4.



Testing the Compass:

The compass sensor is very sensitive, and can easily be affected by magnetic fields or defects — much like a normal compass. This is the reason that our attachment holds the sensor high above the robot — out of the (small) magnetic field around the NXT, and also out of harm's way. Let's test the sensor to see how it works.

- Open the NXT Programming software and create a new program called CompassTesting.
- Find the compass block you've imported, or if you haven't imported it, do so now.
- Drag the block into your program, and click the tab in the bottom left corner of the block to expand all

of its data ports.

- Mouse over the data ports to get an idea of what each one is. If you want to learn more about their function, you can visit HiTechnic's website and look at the documentation.

Now let's write a simple test program.

Program 1 Instructions

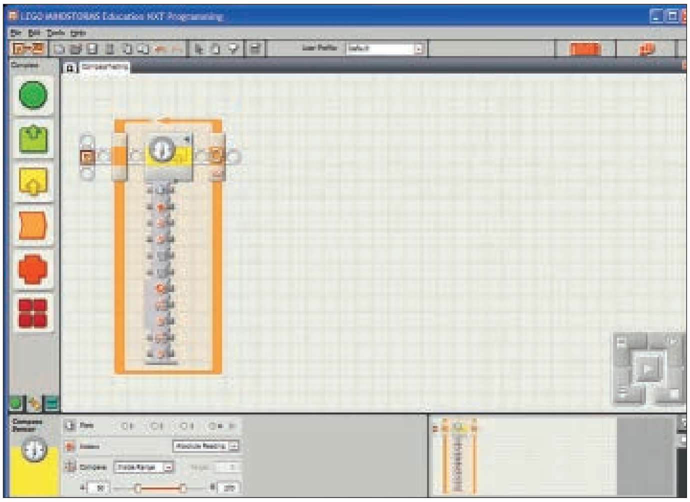


Figure 1. Start by making sure your compass sensor is set to port 4. Add a loop to your program and drag your compass sensor block into it.

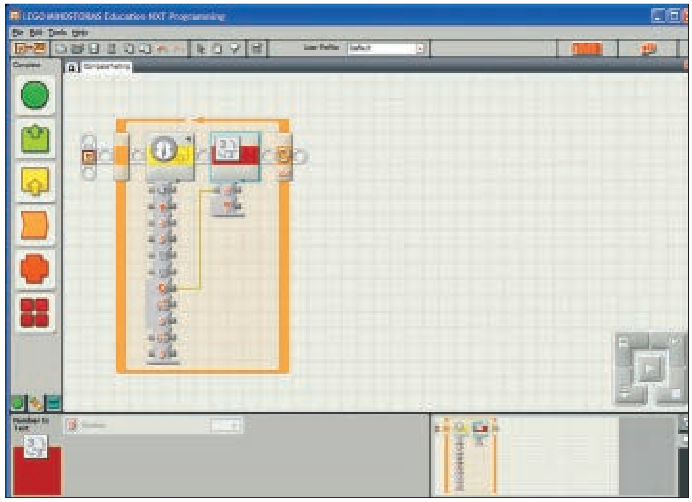


Figure 2. From the Advanced palette, find the "Number to Text" block and add it after the compass sensor. Wire "Absolute Heading" on the compass sensor to "Number" on the Number to Text block.

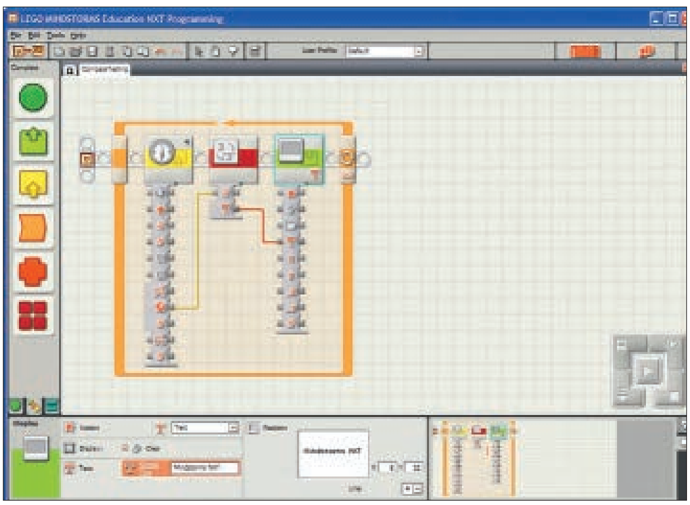


Figure 3. Add a display block from the common palette. For Action, select Text. Then, wire the output from the Number to Text block to the Text data hub on the display block.

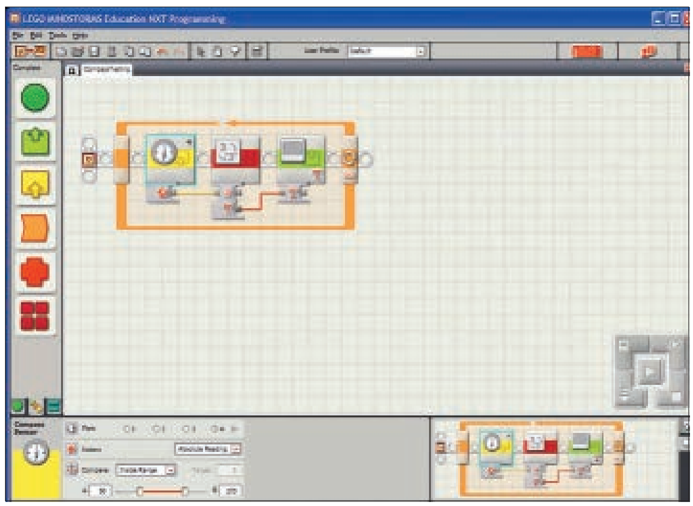


Figure 4. Finally, click the tabs to collapse them and make your program easier to read. Now, download and run your program.

If all goes well, Eddie should now display a number between 0 and 359 on his screen. Notice that the closer Eddie's compass is to pointing North, the closer the numbers will be to 0 or 359. This is Eddie's absolute heading; 0 is North, 90 is West, 180 is South, and 270 is East.

This is awesome, because using this information we can make Eddie go in a constant direction, opening the door to endless possibilities! Let's try getting Eddie to always steer North. Follow these steps to get Eddie moving.

Eddie Faces:	He Should Turn:	Compass Value	Steering Value
Due North	N/A	180	0
Due East	Left	270	-100
Due West	Right	90	100

Because Eddie's compass sensor faces backwards, it reads 180 when he's pointing due North. Remember: Steering needs to be a number between -100 and 100, where -100 is a full left turn, 0 is straight forward, and 100 is a full right turn. Let's put together a table of values to figure out what our formula should be to make Eddie go North.

To turn 180 into 0, you subtract 180. Subtracting 180 from 270 gets you 90, and multiplying 90 by -1 gets you -90 which is close enough to -100 for our purposes.

This means we'll want to subtract 180 from our value, then multiply it by -1. Let's do a final test to see if it will work with West.

(90) - 180 = -90
-90 * -1 = 90

Perfect! Now let's get this in a program!

Program 2 Instructions

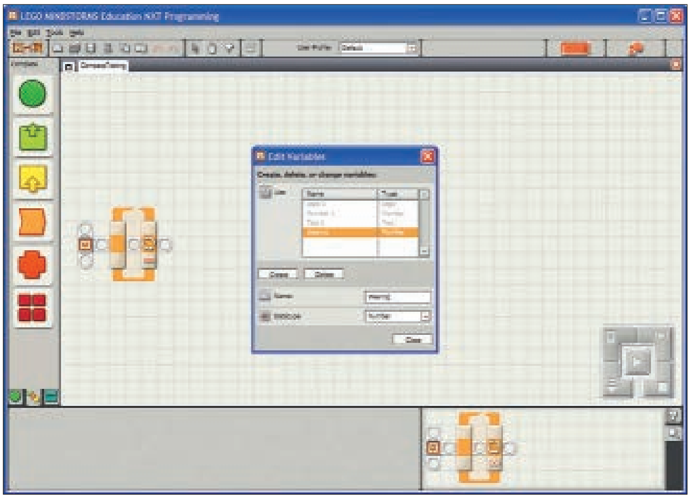


Figure 1. Start with just a loop. Click Edit > Define Variables and create a new Number variable called Steering.

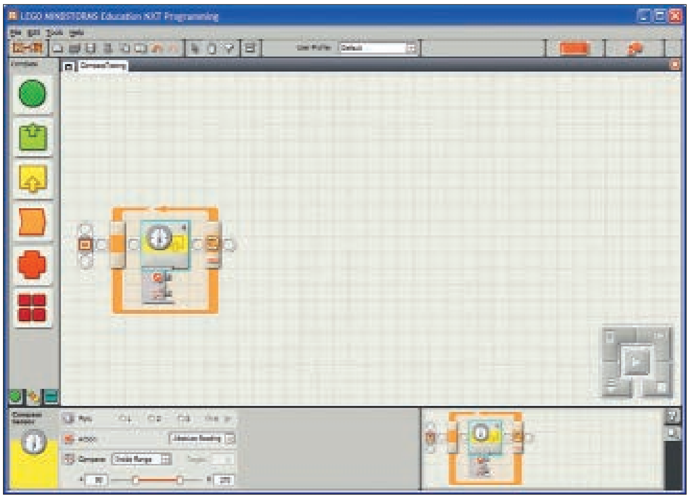


Figure 2. Add a compass sensor block, and set it to port 4.

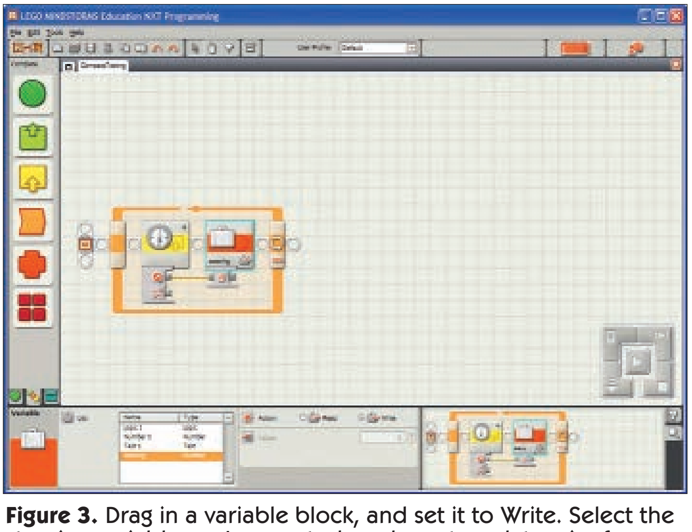


Figure 3. Drag in a variable block, and set it to Write. Select the steering variable you've created, and create a data wire from the Absolute Heading hub on the compass sensor block to the variable input.

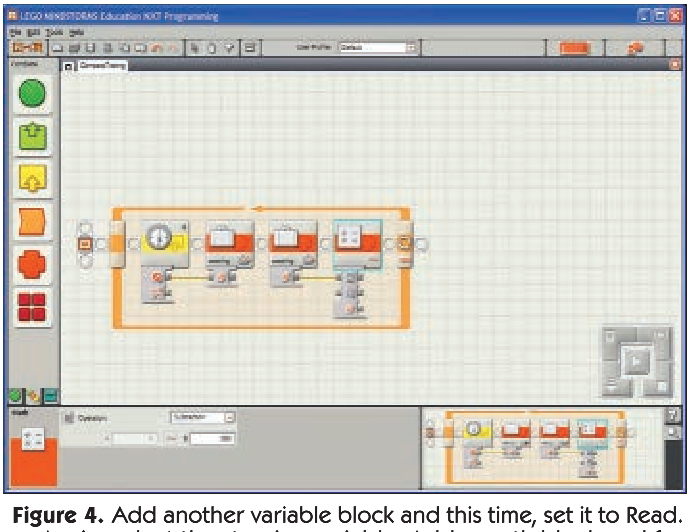


Figure 4. Add another variable block and this time, set it to Read. Again, select the steering variable. Add a math block and for operation, select subtraction. Finally, set B to 180 and put a data wire from the steering variable to the A hub on the math block.

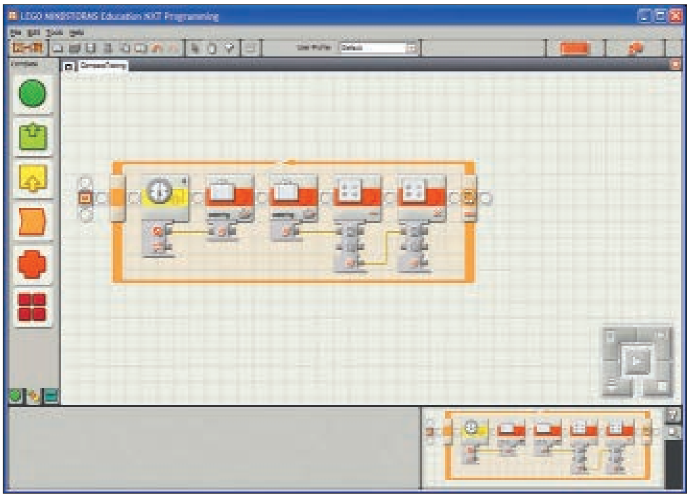


Figure 5. Add another math block set to multiplication. Run a wire from the output of the subtraction block to A on the multiplication block. Set B to -1. This will reverse our values, making Eddie turn in the right direction.

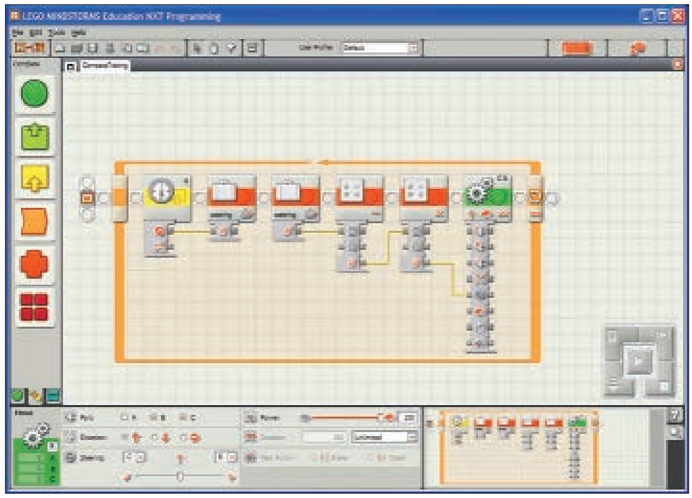


Figure 6. Add a motor block. Set B and C forward at full power, and run a data wire from the output of the multiplication block to the steering data hub on the motor block.

Test Eddie out. He should naturally turn North. If you're having problems, double-check that your left motor is plugged into port C and your right into port B.

Awesome job! You just used a compass sensor to make a robot continuously move North! In the next edition of The NXT Big Thing, we'll take it one step further and push the compass sensor to the limit! Stay tuned! **SV**

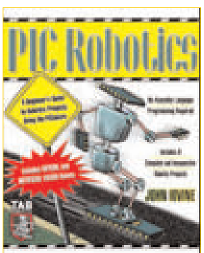
The SERVO Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

ROBOTICS

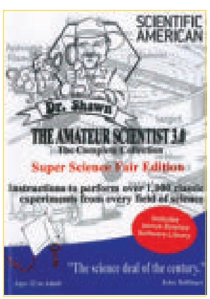
PIC Robotics
by John Iovine



Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU!

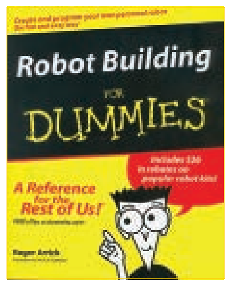
In this heavily-illustrated resource, author John Iovine provides plans and complete parts lists for 11 easy-to-build robots each with a PICMicro "brain." The expertly written coverage of the PIC Basic Computer makes programming a snap – and lots of fun.
$24.95

The Amateur Scientist 4.0 The Complete Collection
by Bright Science, LLC



There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do your first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!
Reg $26.95 Sale Price $23.95

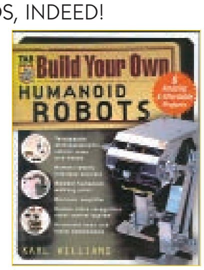
Robot Building for Dummies
by Roger Arrick / Nancy Stevenson



Discover what robots can do and how they work. Find out how to build your own robot and program it to perform tasks. Ready to enter the robot world? This book is your passport! It walks you through building your very own little metal assistant from a kit, dressing it up, giving it a brain, programming it to do things, even making it talk. Along the way, you'll gather some tidbits about robot history, enthusiasts' groups, and more.
$24.95

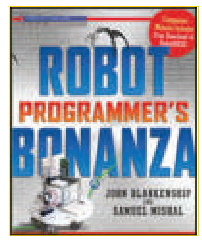
Build Your Own Humanoid Robots
by Karl Williams

GREAT 'DROIDS, INDEED!



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot. **$24.95***

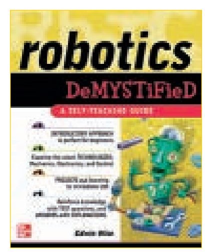
Robot Programmer's Bonanza
by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.
$29.95

Robotics Demystified
by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

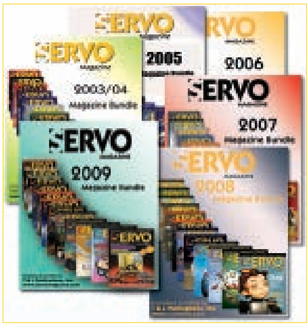


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



Published by T & L Publications, Inc.

$57
per bundle

Save $10 off the normal price!!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, and 09.

Call my Webstore and you'll get someone in AMERICA!

Visit my online store @
www.servomagazine.com

Kickin' Bot

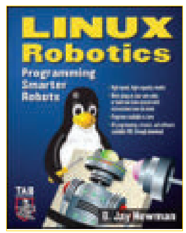
by Grant Imahara

Enter the arena of the metal gladiators!

Do you have what it takes to build a battle-ready robot? You do now! Here are the plans, step-by-step directions, and expert advice that will put you in competition — while you have a heck of a lot of fun getting there. Grant Imahara, the creator of the popular BattleBot Deadblow, shares everything he's learned about robot design, tools, and techniques for metal working and the parts you need and where to get them.
$24.95

Linux Robotics

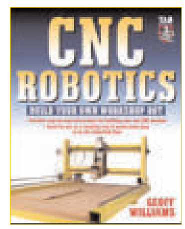
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.
$34.95

CNC Robotics

by Geoff Williams

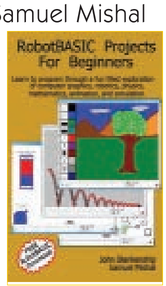


Here's the FIRST book to offer step-by-step guidelines that walk the reader through the entire process of building a CNC (Computer Numerical Control) machine from start to finish. Using inexpensive, off-the-shelf parts, readers can build CNC machines with true industrial shop applications such as machining, routing, and cutting — at a fraction of what it would cost to purchase one. Great for anyone who wants to automate a task in their home shop or small business. **$34.95**

SPECIAL OFFERS

RobotBASIC Projects For Beginners

by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.
Reg. Price $14.95 Sale Price $9.95

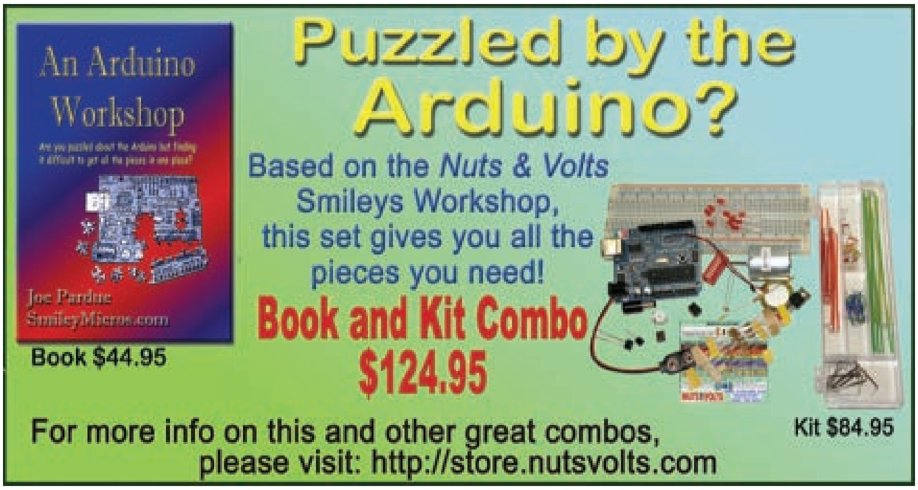
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $79.95 **Subscriber's Price $75.95**

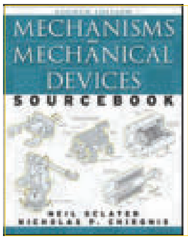
Or order online
www.servomagazine.com

SPECIAL OFFERS



Mechanisms and Mechanical Devices Sourcebook

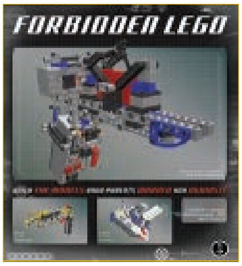
by Neil Sclater, Nicholas Chironis



Over 2,000 drawings make this sourcebook a gold mine of information for learning and innovating in mechanical design. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology will get you up to speed on these cutting-edge technologies. Easy-to-read tutorial chapters on the basics of mechanisms and motion control will introduce those subjects to you. **Reg $89.95** **Sale Price $69.95**

Forbidden LEGO

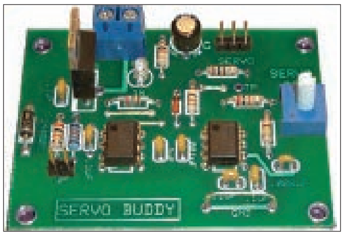
by Ulrik Pilegaard / Mike Dooley



Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **Reg $24.95** **Sale Price $19.95**

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

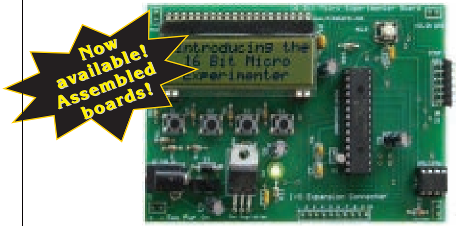
For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

16-Bit Micro Experimenter Board

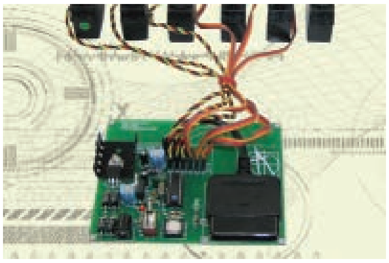


Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck! In the December 2009 *Nuts & Volts* issue, you're introduced to the 16-Bit Micro Experimenter. The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in upcoming months.

Subscriber's Price **$55.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

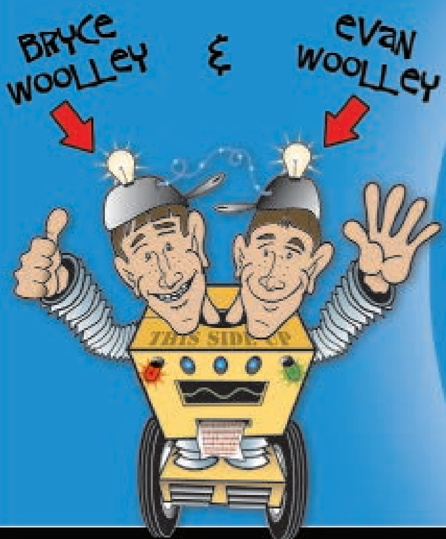
For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

Twin Tweaks

Bryce Woolley & Evan Woolley



THIS MONTH:

Little Robot Shop Of Adventure

The DF Rover from RobotShop and DF Robot.

Arduino — the open source prototyping platform — has undeniably taken the robotics world by storm. The storm has finally reached our corner of *SERVO Magazine*, and we are excited to recount our adventures with the DF RobotShop Rover. The DF RobotShop Rover is a collaborative effort between developer DF Robot and distributor RobotShop. The Rover is an Arduino "clone" meant to be useable as a platform more readily supportive for robotics projects because it includes a motor controller and two small DC motors, in addition to all of the hallmarks of the Arduino board. Such a platform is as surefire a springboard for adventure as finding a map on the back of the Declaration of Independence, and we were excited to embark on an exploration of the Rover's capabilities.

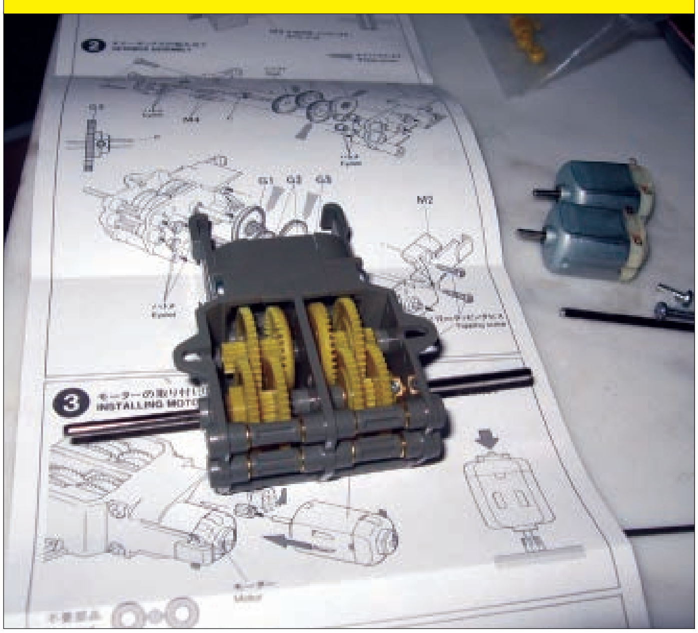
International Treasure

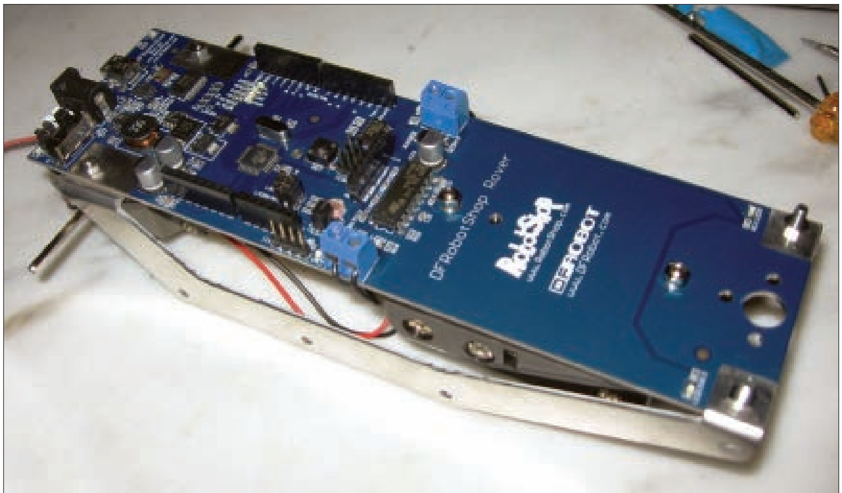
Arduino, along with delicious cuisine, Renaissance art, and Top Chef's Fabio Viviani has taken its place as one of Italy's great cultural exports. Developed by Arduino Software, the Arduino is an open source prototyping platform that consists of (at its core) an Atmel AVR microcontroller, an H-bridge for voltage control, and a voltage regulator to keep all of your electronic add-ons from burning out. Several fantastic articles from our colleagues at *SERVO* have made an in-depth investigation of the technical specifications of the board for those hungry for more statistics and details (see the Nov '10, Dec '10, and Jan '11 issues for some excellent Arduino coverage).

The RobotShop DF Rover, however, is not actually an Arduino robot per se. The Rover is one of many Arduino

Attack of the Arduino clone — the main PCB of the Rover.

Assembling the Tamiya gearbox.





Building up the Rover base.

clones – kits not designed by the folks at Arduino, but containing the same basic elements and compatible with the Arduino software. The most basic of those basic elements is the ATmega328 — the same microcontroller as the Arduino Duemilanove board.

The Rover also uses the same basic header configuration as an Arduino board, with sets of sockets for analog and digital input and output. Far from brazen copying, the existence of such a clone is one of the major goals of the Arduino project. As an open source platform, the folks at Arduino encourage the development of clones and compatible third-party offerings.

The Rover kit can actually be purchased with some of those very third-party offerings. The basic Rover kit comes with the Rover PCB, the Tamiya gearbox and motors, and the Tamiya tracks. RobotShop also offers an Xbee kit, complete with the aforementioned basics plus an Xbee shield, XBee modules, and XBee USB interface. The basic kit is programmed using a USB cable, but the XBee kit allows for wireless programming and control. RobotShop also offers a Bluetooth kit which is what we received. The Bluetooth kit similarly comes with a Bluetooth shield and USB interface.

In addition to the Bluetooth shield, we were given a sampling of some of the many shields available for the Rover and any other Arduino compatible board. Shields — for the uninitiated — are simply PCBs that connect to the headers on the Arduino board. Shields contain the same headers for digital and analog I/O as the Arduino itself, and they include some additional functionality. Our kit came with two such additional shields, other than the Bluetooth module. One was the Solar Charger from Seeedstudio. This allows users to charge a rechargeable battery pack with a tiny solar panel.

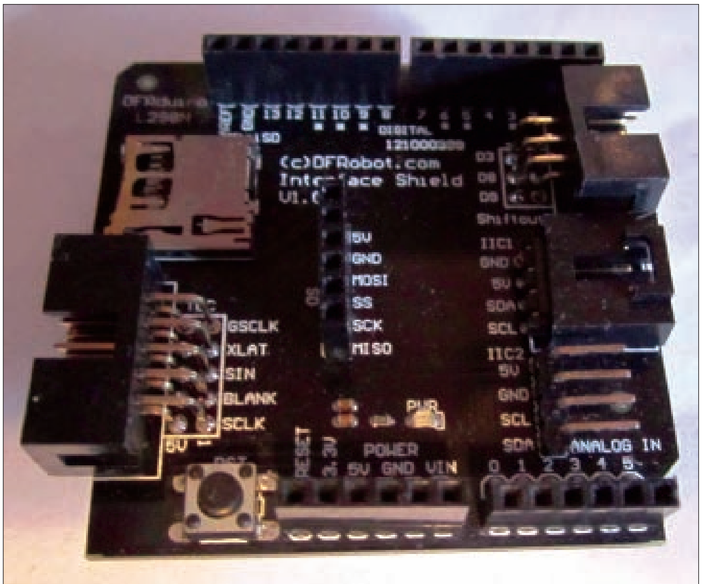
The second was the Interface Shield from DF Robot. The interface shield comes with a plethora of ways to implement an LCD screen. The interface shield includes headers for SPI and IIC connectivity, and even includes a spot for a mini SD card. And, if all of that sounds impressive, some intrepid Arduino users have cataloged up to 219 different shields from developers all over the world. The possibilities were dizzying, but even the most epic of Arduino quests begins with the first step. For us, that first step would be to assemble the Rover.

The Arduino Code

The Rover kit comes with no attendant CD, because everything is available online. The RobotShop website offers an instruction manual which gives step-by-step instructions for building the kit. At first glance, the instructions looked positively anemic – each step was about a line of unadorned text, devoid of any helpful diagrams. However, a second more than cursory glance revealed the suggestion of watching the assembly video on the website. The video on the website gives helpful and visually detailed instructions for building the Rover – everything from assembling the gearbox to installing the treads. Instructions are written onto the screen, and a rockin' techno soundtrack evoked the slick stylishness of a CSI labwork montage.

The first step was to assemble the Tamiya gearbox. The gearbox can be assembled in three different configurations: one that maximizes torque; one that maximizes speed; and one that strikes a balance. The narrow track of the Rover requires the narrower configuration, designated C. C was also the torque maximizing configuration which seemed like a good choice given our desire to load up the Rover with mechanisms and sensors. The gearbox — which does come with its own instructions — goes together smoothly. The kit includes the small Allen wrench necessary to tighten the set

The Solar Charger shield from Seeedstudio.



THE INTERFACE SHIELD FROM DF ROBOT.

screws on the shaft, and the only other tool needed is a small Phillips screwdriver.

The Rover prides itself as a solderless kit, and we think this is indeed a great way to entice newcomers to dive into the vast universe of Arduino. The only part of the basic kit that could possibly require any training in the art of metallurgy would be in attaching wires to the DC motor leads. The Rover, however, makes good on its promise of no soldering by offering leads that can be crimped onto the motors instead. We preferred soldering, but the solderless option was an early and encouraging sign of accessibility.

The only other major assembly was with the frame and treads. The minimalist frame uses two aluminum supports attached with a total of four screws. The gearbox attaches to the PCB by picking up the two mounting points in the back of the robot. The only tools required are a Phillips screwdriver and some dexterous fingers. The treads are also from Tamiya, and even though the sprockets simply press onto the axels, they provide a solid way to drive the Rover.

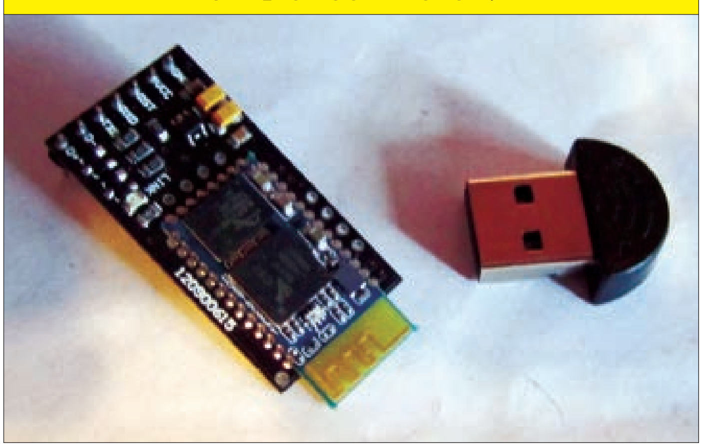
The next step was to program the Rover. The Arduino software is available for free on the Arduino website. Just like the Arduino board itself, the Integrated Development Environment (IDE) is simple, sleek, and a pleasing shade of blue. The IDE is actually based on the open source Processing language which was designed to introduce novice programmers to the joys of coding in a friendly and visual way. Arduino builds on the syntax and libraries of Wiring — an open source sister project to Processing meant to apply Processing to microcontrollers.

Programs in the Arduino software are called "sketches," and the IDE uses syntax highlighting, brace matching, and automatic indentation to shepherd the novice programmer. More seasoned programmers — particularly those fond of C/C++ — will feel right at home doodling away in their "sketchbook." The software, much like the Arduino itself, is being consistently improved and it is regularly updated, so much so that during the time of our project the software was upgraded from version 0021 to 0022. All of the previous versions of the IDE are also available, and the truly cosmopolitan folks at Arduino offer packages compatible with Windows, Mac OSX, and Linux 32- and 64-bit operating systems.

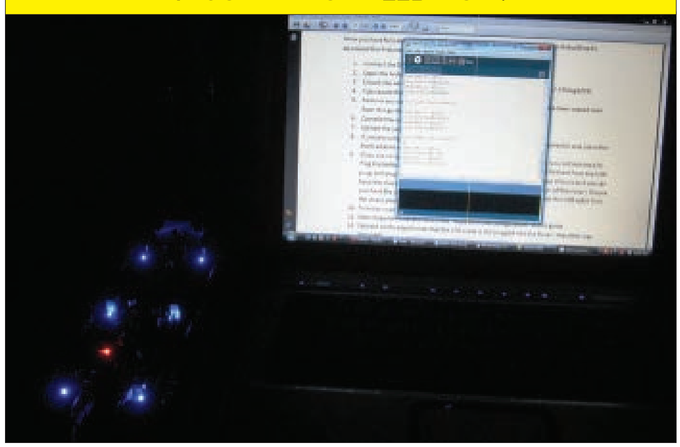
The manual provides step-by-step instructions for firing up the Rover for the first time and writing your first program. The first program that the manual has in mind allows the Rover to be controlled with the computer either wired with a USB cable or wirelessly using XBee or Bluetooth modules. The movements of the robot are commanded by specific keystrokes. In an ingeniously user-friendly move, the manual offers this simple beginning program as something that can be copied and pasted directly out of the manual itself.

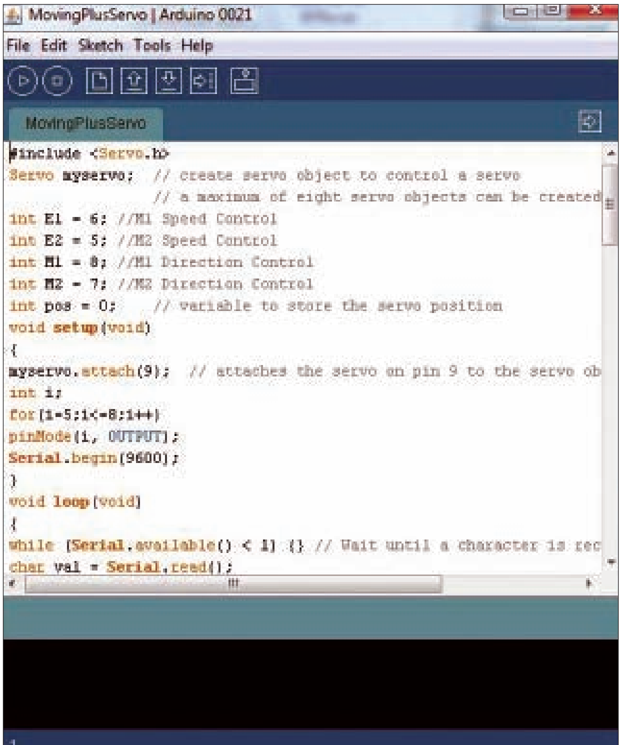
Upon hooking up the Rover to the computer via USB, we were greeted with the welcome sight of LEDs beaming to life. LEDs on the robot can be adjusted with jumpers, and they can be used to indicate the direction of the motors. Downloading the program was a refreshingly easy task that presented no difficulties whatsoever, but there was one more step before we could run the robot from the computer.

THE BLUETOOTH MODULE.



PROGRAMMING BY LED LIGHT.





THE ARDUINO IDE, BASED ON THE PROCESSING LANGUAGE.

HACKING THE CONNECTOR.

Operating the rover with the sample program requires communicating with the bot through HyperTerminal which comes as a standard feature on windows XP and 2000, and unfortunately does not come on the more recent Windows Vista and Windows 7. We were able to just do a quick Google search and download a 30 day trial from Hilgraeve — the folks behind the very program that comes bundled with older versions of Windows. There are also numerous other emulators available. The hassle of tracking down a HyperTerminal program is worth it because the window will be the space where infinitely helpful information can be printed by the program. After defining a connection in the HyperTerminal program, the Rover was roving steadily at the strike of a key.

That Will Lead to Another Hack, and That Will Lead to Another Hack!

The basic movement program is great fun to play around with. The robot is easy to control, and the high torque gearbox ensures that it zips along at a manageable pace while being able to climb over all sorts of terrain with its angled treads. Because the robot was taking input from the computer, it had to be hooked up with a USB cable (thankfully, it's an easy to replace Mini B type). We hated to see the wild bot tethered, so we thought that a good first step in testing the expandability of the Rover would be to implement the Bluetooth module. The basic Rover instruction manual gives step-by-step instructions for just such a task, complete with screenshots. Physically connecting the module was easy — the Rover has an extra set of headers compared to the actual Arduino boards; one specifically for wireless communication modules. The only stumbling block was that the Bluetooth module does not come with its own software. We ended up shelling out $20 for the recommended software from Blue Soliel, but it was user friendly and compatible with a broad range of USB devices.

Establishing the Bluetooth connection takes a few more steps than the wired connection, and the first few times we tried, the connection would die before the program had completely downloaded. After checking the COM port and the HyperTerminal connection, we were eventually able to operate the robot wirelessly — all without soldering or changing the program.

After our initial successes, we were eager to add even more to the Rover. Coming from a mechanical engineering background, we have always appreciated the subtle artistry of the good old fashioned claw, and we thought such an addition would be quite necessary because one does not simply rove into Mordor. For the claw, we scrounged up a hardy Futaba servo. The servo had a 180 degree rotation which would be more than enough for our purposes. The front of the robot was the logical place for the servo, though we were a bit nervous about the balance of the bot. The front was also the place with the best mounting options with both the universal mounting point and the holes for the frame. We thought to defer our balance fears until after we wired up the addition.

All of the headers on the Rover are female. Unfortunately, it seemed like every addition that we wanted to make to the Rover also sported female connectors. There are adapters available, but that would add extra length to wires already more than long enough for the kit. Thankfully,

we had some MilSpec connectors on hand that were a perfect source of pins that we could hack onto the cables. Given the proper tools, such a hack would have been exceedingly simple. We had the tool to remove the pins from the connector itself, and the quickest way to attach the pins to our new cables would have been to crimp them on. Unfortunately, we did not have the right size crimper, but all was not lost. Some solder and heat shrink saved the day.

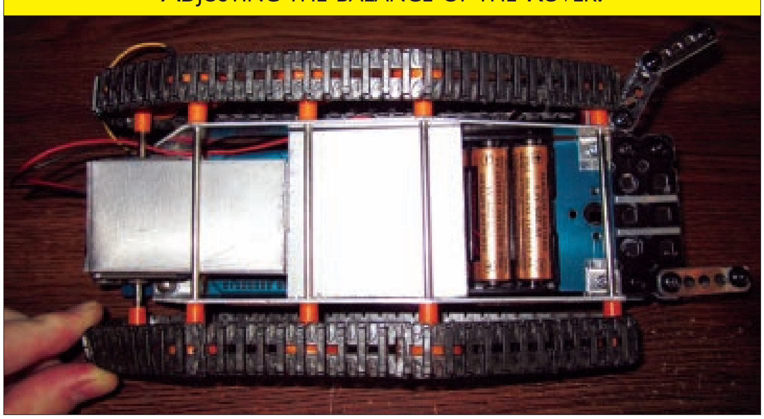
Looking at the Rover PCB, we could see analog ports and we could see digital ports, but we wanted to connect our claw to a PWM port. A quick reference to the website revealed the answer to the riddle — several of the digital ports had PWM capability, and the talented ports were labeled with an asterisk (they are labeled as PWM on actual Arduino boards). A sample program that came with the Arduino software was called Sweep and would move a servo back and forth through a 180 degree rotation. We modified the code to have a good grasping range, and one quick download later we were greeted with the welcome sound of a servo whirring to life.

Our claw, unfortunately, was too tall and too heavy, and it moved the center of gravity beyond the tracked wheelbase. To fix the problem, we wanted to add weight to move the center of gravity back safely over the wheelbase. The Rover, however, did not have a lot of other structural attachment points. Other than the universal mounting point, the only other convenient place to mount attachments would be in the holes meant for the battery holder or tread frames.

Time might be wisely invested in devising a custom frame for the Rover — perhaps something that looks like the frame for the treads that can attach to the same holes, but extends around the top of the robot like a roll cage.

ADDING A CLAW AND SHARP RANGEFINDER.

ADJUSTING THE BALANCE OF THE ROVER.



Crossbars on the new frame could act as shelves for ballast or additional devices. The screws on the front mounting points are long enough to accommodate additional frame bits, but the screws in the back — which already hold the frame, PCB, and Tamiya gearbox — don't have the ability to hold anything else. Of course, correcting this would simply be a matter of picking up some longer screws, but we were looking for a fix that we could execute without trekking to the hardware store.

Turning the Rover over, we were greeted by the sight of the smooth axles that supported the idlers for the treads. The axles spun freely of the idlers, so we thought that we could actually use them as a sort of basket to hold ballast. Fortunately, Robot Central is still a fountain of useful aluminum scraps, and we found some beefy half inch thick slabs that were sure to counteract the tipsy effects of our claw. We cut the aluminum down to size, and the half inch thickness was the perfect snug fit between the axles and the battery pack. To ensure that our Rover wouldn't pull a Johnny 5, we wrapped the ballast in duct tape as an insulator. We also slid in layers of thinner aluminum between the axles and the Tamiya gearbox for some more weight at the very back of the robot. The ballast restored the balance to the Rover, and once again allowed it to tackle tough terrain.

The claw, however, was not very useful when the Rover had no way of knowing what to grab on to. Fortuitously, the Rover came with a Sharp GP2D12 infrared rangefinder. The useful sensor pops up everywhere — like on our bygone project with German robot Crash Bobby. The GP2D12 is a useful analog sensor that gives a voltage reading corresponding to the distance of an object from the sensor. The kit also included a mounting bracket designed to mate with the universal mount on the front of the Rover. Unfortunately, that spot was taken up by the claw, but the bracket was still bound to be useful. The kit also came with a cable for the GP2D12. Once again, however, the connector was not suited to connecting with the headers on the PCB. We still had the soldering iron and heat shrink prepared from the last connector hack, so the GP2D12 was quick work.

The super helpful Arduino website includes the Arduino Playground — a wiki where hackers from around the globe share their insights and experiences. One of the parts that we found most helpful was a comprehensive guide to implementing various sensors. The guide features both readable diagrams for wiring up the sensor and code that can be readily copied

and pasted into your very own sketch. The GP2D12 is a popular sensor, and as expected, that Arduino website had code written by an altruistic roboticist that converted the voltage reading from the sensor into centimeters. We mounted the GP2D12 so that it had a view of the claw, and after messing around with the ideal grabbing range, the claw would close on whatever happened to be in range.

Recommended Websites

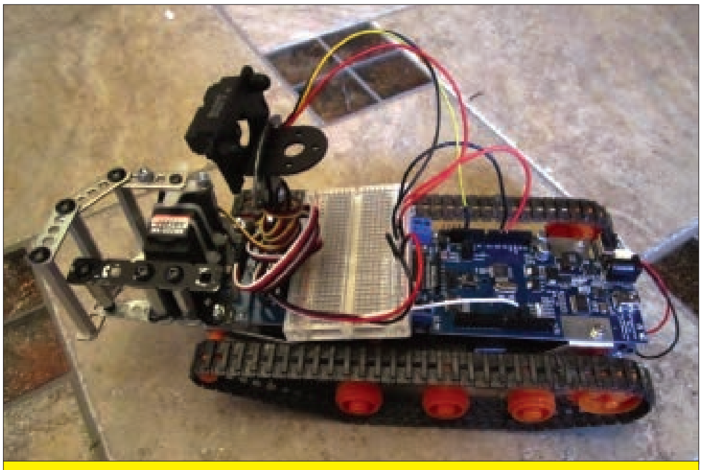
To get your very own Rover!
www.robotshop.com

The official Arduino website
www.arduino.cc

For a staggering list of 219 Arduino compatible shields
http://shieldlist.org

The Rover had plenty of I/O ports to support the servo, rangefinder, and much more. The only difficulty we could see was that the Rover had only two sockets that sourced five volts, and one of them was on the Bluetooth module header. This can be easily solved with a prototyping shield or by using a mini breadboard also available from RobotShop.

It just so happened that brightly colored Easter eggs were particularly easy to identify and grip. Even though it is only March, the hallmark of a good adventurer is preparedness, and sending the Rover on an Easter egg hunt would be an amusing way to test the efficacy of our hacks. While avoiding the entanglement of Easter grass, we were impressed at how effective the Rover was at climbing over terrain even with the additional mechanisms and ballast.



WIRING UP WITH A BREADBOARD.

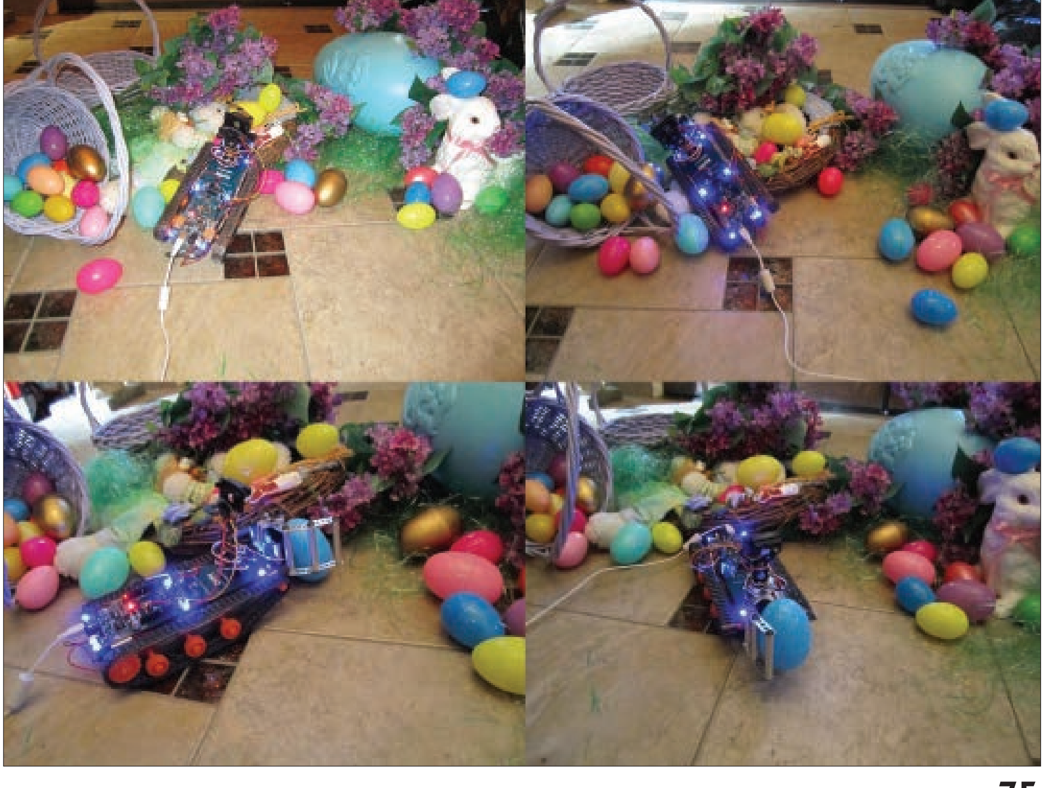
robot is as effortless as building up a LEGO kit. The Rover is even compatible with the Catcan sensor we worked with in the January issue, and the potential for robotics projects is quite literally limited only by your imagination.

The Rover is a wonderfully user-friendly and compulsively expandable kit, and we think its affordability and rich community of online support make it perfect for beginners looking for a good project, while the depth of third party offerings will keep even the most seasoned hackers busy. Our only minor concern is the structural expandability of the kit, but we're sure by the time of this printing that the Arduino Playground will be teeming with creative solutions. SV

Book of Open Secrets

We were pleased with our hacks, and we only barely scratched the surface of the Rover's capabilities. Never before have we used a kit with such thorough and exciting offerings from third parties, and we think a lot of this has to do with the fact that the good folks at Arduino made the board open source. The open source movement has resulted in some great technological innovations ranging from Linux to Minecraft, and we think that Arduino is deservingly taking its place as a worldwide standard for hackability and prototyping. The programming is intuitive, and the physical expansion of the

SEARCH AND RESCUE!



Then and NOW

TELEPRESENCE

by Tom Carroll

Telepresence is a word that is used quite a bit these days as experimenters attempt to add some sort of two-way television system to their robotic creations. The concept is not new and began to gel in the minds of experimentors soon after the invention of the telephone. Just as Alexander Graham Bell and others of his time desired communications with others at a distance, the enhancement of this scenario would certainly be more complete with the addition of a picture of the other person (or persons) at a distance. These days, we have passed the point of video telephones with Cisco, Google Talk, Skype, iChat, and similar systems on our computers, but the steps to these technologies arrived slowly. Far too slowly for many.

Alexander Graham Bell

Before I discuss the many telepresence robots that have been marketed in the past and those available today, I'd like to look into the history of hearing and seeing at a distance, and different ideas on the system's design concepts. We all know of Alexander Graham Bell as the inventor of the telephone that he patented back in 1876. Bell was interested in more than just the telephone. He actually considered his device an intrusive nuisance and never had one in his study.

Born in Scotland in 1847 and already an inventive kid at age 12, he designed and built a corn de-husker for a friend's father who owned a flour mill. Later in life, he worked on many technical devices including a metal detector and hydrofoil boats — one of which held a speed record of 71 MPH for 10 years. He was also interested in manned kites and heavier-than-air powered aircraft back in 1891 before the Wright's first manned flights. With an interest in vision systems, he held patents for the use of selenium cells as electrical light and vision devices.

As with another amazing inventor of the same era — Thomas Edison — Bell was also beset with numerous legal hassles with his patents. Through all his problems with his patents and his illnesses from childhood on, it was his interest in the human voice and hearing that captivated him the most. Moving from Scotland to London, then to Canada, and finally to the Boston area in the US, Bell acquired quite a bit of knowledge and background in dealing with the deaf. It was his mother's slow loss of hearing that led him to the study of acoustics. At age 16, he secured a position of "pupil-teacher" of elocution and music at the Weston House Academy in Scotland. He later worked closely with deaf students and later married one. His interest spanned across speech generation to speech recognition.

In 1875, Bell developed an acoustic telegraph and filed a patent for it in March of that year. He did not realize that a competitor — Elisha Gray — had filed one the month before. Both systems used a 'water transmitter' principle, vaguely similar to the graphite granule transmitters that were developed later. Both used the changing resistance of a diaphragm's pressure and vibrations on the water or graphite. We've all heard of that famous first 'phone call' on March 10, 1876 where Bell called to his assistant, Thomas Watson, from another room into a transmitter with the words: "Watson, come here, I want to see you." Watson supposedly clearly heard the words on the receiver and came to him, though many historians doubt these were the actual words.

Bell Looks Past the Telephone

It is interesting that Bell used the words: "I want to *see*

you." Why not "I need you?" Maybe he was wishing that his new device was a bit closer to what we now call telepresence. Some of Bell's technical notes from April 1891 indicate that he was thinking about an 'electrical radiophone' when he recorded *"... the possibility of seeing by electricity."* He mentioned using "specks of" selenium as part of the vision imaging elements. He wrote: *"Should it be found ...* [that the image sensor] *is illuminated, then an apparatus might be constructed in which each piece of selenium is a mere speck, like the head of a small pin; the smaller the better. The darkened selenium should be placed in a cup-like receiver which can fit over the eye ... Then, when the first selenium speck is presented to an illuminated object, it may be possible that the eye in the darkened receiver should perceive, not merely light, but an image of the object... "*

It was easy to see that light-sensing devices developed from using these elements could send an electrical signal proportional to the amount of light shining upon it. However, a complex image was beyond the capabilities of these sensors. Even the crudest image would require many hundreds of separate pixels or specks of selenium to sense one scene and form that scene at another location. He had no idea of how to address each line of specks and how to receive the varying electrical signals from each of the specks, then transmit them, and then re-arrange them at the other end into a picture. Would he have to divide each image into a line of pixels and then scan each line individually or does each illuminated pixel need to send a bit of information to its respective pixel at the other end?

With the success of his telephone that transmitted verbal information, Bell predicted that: *"... the day would come when the man at the telephone would be able to see the distant person to whom he was speaking."* **Figure 1** shows an artist's conception of 'Videotelephony' as imagined in 1910 — 20 years after Bell's prediction.

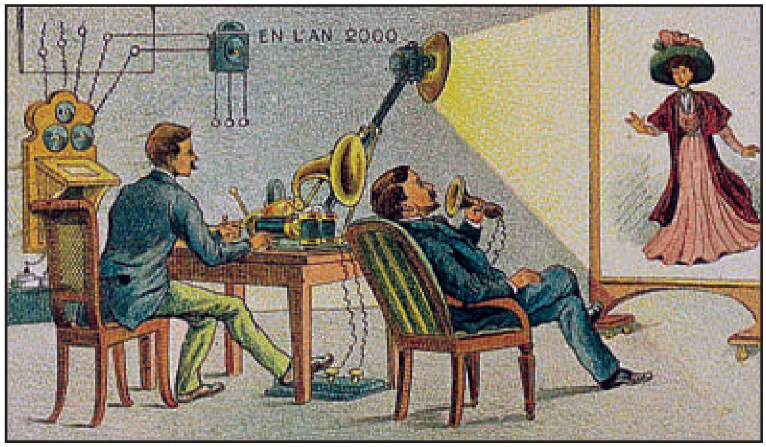


FIGURE 1. Artist's conception of Videotelephony, as imagined in 1910.

Television Makes the Scene

Basic telepresence got its true start with the invention of television. When the cameras and monitors were first available, inventors immediately interconnected them in pairs to provide remote two-way television communications. Of course, thoughts about developing the viewing and transmission of visible images started back in the 19th century with Bell's and other's ideas — well before what we now call television. The telephonoscope was first envisaged in 1878 with some crude sketches by science fiction writers. The idea of scanning the image used a swinging pendulum in 1881, but the best idea of the era was using a spinning disc with a series of holes in it, spiraling towards the center to create a raster scan. With the development of vacuum tube amplifiers and other circuitry, radio wave transmission, and the cathode ray tube (CRT), television became one of the main means of home entertainment.

Regular television transmissions began in 1929, and the 1939 New York World's Fair introduced it to the American public. World War II kept wide acceptance and availability from the public until 1948. Color television appeared in the NTSC form in the US in 1953. TV resolution in the US started below 300 lines of scanning for the picture, but soon standardized at 525 lines. Most European countries used the higher 625 lines with 30 frames per second interlaced to provide 60 lines. Solid-state electronics, LCD, projection sets, and high definition brought us to where we are in this new millennium.

Telepresence Starts with Video Phones

With the transmission of audio and video over wire and RF, Bell's wish for true telepresence was possible, but not quite to the extent that we have it in 2011. I remember seeing a video phone at Disneyland when I was a kid and

FIGURE 2. The Jetson's video phone.

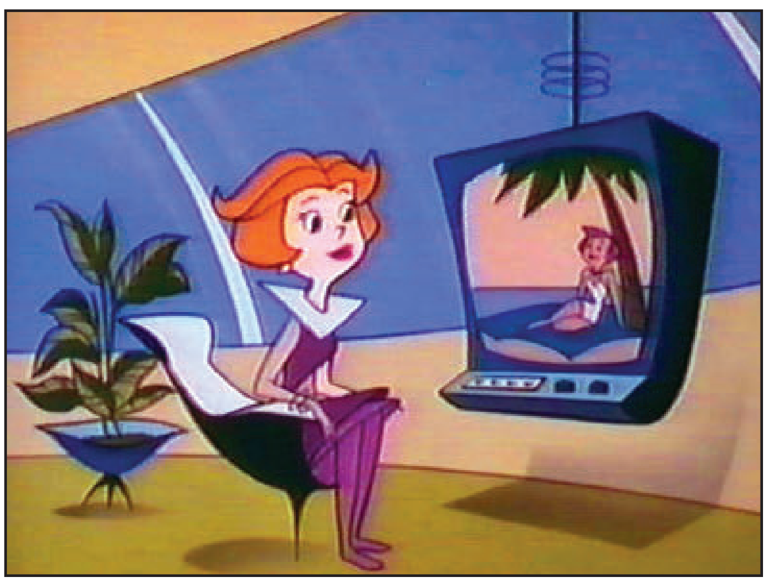


FIGURE 4. Nortel video phone.

FIGURE 3. Artist's conception of a video phone in the 1950's.

thinking that such a thing was so cool that every home would have one soon. (Hey, if the Jetson's could have one <**Figure 2**>, surely engineers could connect a TV and a two-way radio together so we could talk with friends and *see* them too.) TVs in the '60s were all tube powered and were quite large. The smallest 'portable' sets were the size and weight of a large tool box, and small color sets were too complex to make in small sizes. Although, the 1950's artist's conception in **Figure 3** showed what an office of the future might look like.

CRTs were the only image-producing devices and were usually quite long for the small screen sizes, but that didn't stop companies from producing small video phones. Small CRTs that utilized a 90 degree bend for the electron gun could be placed in a small package that allowed the video phone to be mounted on a wall. AT&T debuted its *Picturephone* at the 1964 World's Fair in New York and at Disneyland, but it later bombed in the market. They tried again in the early '90s with a newer model priced from $1,000 to $1,500, but still had no success. **Figure 4** is a desk-top office video phone from Nortel. **Figure 5** shows a prop of a video phone used in the 1966 movie, *2001: A Space Odyssey* which the actor used to call down to earth to wish his daughter a happy birthday.

Is Video Calling and Conferencing Finally Accepted?

Most of you probably don't know of a person who has a manufactured video phone in his or her office or home. They just didn't take off as we might have thought that they would. Why? Were we afraid that we'd have to throw on a bathrobe to answer the phone? Cisco and other manufacturers have found great acceptance of video conference calling or teleconferencing where single people or groups can interface in real time from anywhere on the planet. Systems like those in **Figure 6** are usually set up in a dedicated office or conference room with a large screen at one end, and one or more TV cameras to capture views of the room and participants.

Large corporations can afford real time video with high definition and no stop-motion video. Many home systems, however, may have to deal with jerky stop-motion video. The

FIGURE 5. A video phone call from Earth Orbit in the film *2001: A Space Odyssey*.

webcam has enabled Skype and similar technologies in homes and businesses. High quality imaging devices — both on the camera and on the display end of things — are available today at amazingly low cost. DSL and other media has brought high bandwidth Internet to everyone. The next step is to make a telepresence system mobile — such as wih a telepresence *robot*.

I have been watching the development of telepresence robots for several decades. Everyone seems to believe that their machine is the one to beat all the others. Many people really like the idea of a telepresence system that can move itself from one location to another, controlled by a person at another location, but not all feel that the extra cost is worth it. From a robotics enthusiast's point of view, *of course* a mobile robotic base is by far more useful. But is it, really? Just as that webcam embedded into our laptop or sitting atop our monitor can be intrusive into our lives without us knowing it, a mobile camera system in a telepresence robot that can be remotely controlled into a location where it shouldn't be is equally intrusive.

FIGURE 6. Video conferencing Cisco telepresence.

Telepresence in the Form of Texai

When visiting *Willow Garage* in Menlo Park, CA last year, I was shown through much of this unique company's lab spaces. One of the most unique displays at their facility was a line of telepresence robots in a hallway shown in **Figure 7**. This row of Texai (plural of Texas) robots was developed by Dallas Goecker, an EE engineer who lives in Indiana and telecommutes to Willow Garage via *telerobotics* — the new term.

Willow Garage is a most exceptional company that has developed robotic hardware that is second to none. I've written about the PR-2 robot on several occasions, but the Texai robots have recently taken center stage in the media. As a valuable contributor and member to the WG team — and not willing to move from Indiana — Goecker felt it necessary to build an avatar of himself in the form of a robot in order to feel a part of things from thousands of miles to the west. For the base of his first version of the Texas 'Alpha' robot (shown in **Figure 8**), he and fellow engineer Curt Meyers used the motorized castors from a PR-2; the rest was strictly off-the-shelf parts. One of its best features — in my opinion — is their use of WG's powerful open-source ROS (Robot Operating System) that operates teleoperation, camera movement, steering, and

FIGURE 7. A row of Texai — Texas Alphas at Willow Garage.

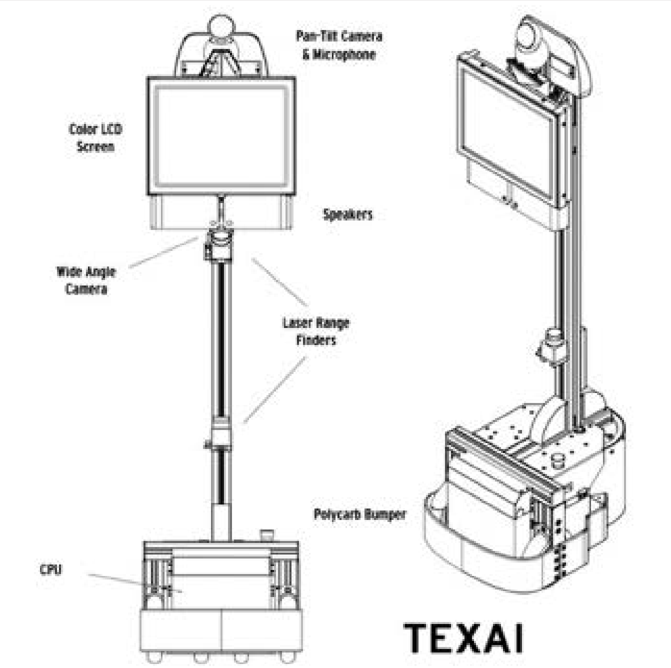


FIGURE 8. Breakdown of the WG Texai robot.

FIGURE 9. The Anybots QA telepresence robot.

wheel driver systems.

Skype is the two-way video link system used with the Texai. A web page allows the remote user to guide the robot using a mouse to position a red dot for directional control. A separate camera looks downward so the remote user can see the base and avoid close-by obstacles such as furniture and people's feet. Even though the Texai are only proof-of-concept vehicles, Willow Garage furnishes new users with a video to teach them how to operate the robot and what not to do when guiding it. I found that the large LCD monitor and the dual axis TV camera set these prototypes apart from any of the competition, as the large image of a person's face closely approximated a real human being in my presence, and made it easier for the individual on the other end to visualize the distant scene and control the robot. Willow Garage has not set a price on these proof-of-concept machines, but newer units use an easier to control differential drive system, rather than the very expensive PR-2 motorized castors that employ modified Ackermann steering.

FIGURE 10. The Anybots QB telepresence robot.

FIGURE 11. The affordable Vgo telepresence robot.

Anybots QB

I've concentrated on the Texas telepresence systems from Willow Garage as I am most familiar with them but there are several more that have also made headlines, including the Anybots QB avatar telepresence robot made in Mountain View, CA (not far from Willow Garage). This startup may not have produced the first telepresence robot but just might be the first to bring one to market. They debuted the QA model shown in **Figure 9** at the Las Vegas Consumer Electronics Show (CES) in Jan '09. At $15,000 a pop, the newer two-wheeled 'Segway' styled second generation QB robot shown in **Figure 10** has an extensible neck and weighs only 30 pounds. It comes with an eight-hour battery life and is perfect to be folded up and carried around to meetings.

Vgo

Vgo Communications is based in Nashua, MA near Boston and brought forth their Vgo robotic telepresence robot in mid 2010. Founded by Tim Root, Grinnell More, and Tom Ryden, the company feels that the $5,000 robot is a perfect solution for today's communications solutions for real world problems — whether that's in the office, home, or hospital environment.

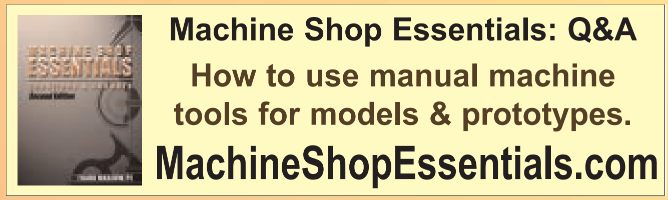
Though the robot shown in **Figure 11** has a mandatory $1,200 per year support contract, Vgo management feels that the initial lower cost is one of the machine's best features. The four foot robot weighs just 18 pounds and has a six to 10 hour operating time.

Final Thoughts

A dozen or so companies are betting that teleconferencing can save companies thousands of dollars over the logistics and travel costs of face to face meetings. A dozen more are betting that telepresence robots will hold an edge over teleconferencing with the personal touch of a remote person's avatar. It's a sure bet that most of us are pulling for the robots. When the true look and feel of a real humanoid robot has conquered the 'uncanny valley' of realism and the individual sitting beside you in a conversation is impossible to determine to be a robot or human, then we've solved the dilemma ... or have we? Is it the robotic concept that appeals most to us or is it the telepresence aspects? Readers are encouraged to share their telepresence projects. SV

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS



For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

To Advertise: Call 951-371-8497

Advertiser Index


robotshop.com/eu
CANADA
USA
EUROPE
RD-1
Now Serving Europe
with Optimized Logistics
Currency in EURO
3 day shipping to 70% of the territory
Service in French and English
Competitive shipping rates
Huge product selection
One Global Door for Manufacturers
VISIT:
Shipping Worldwide
www.RobotShop.com
Robotics at your service! ™

I2I Controls

by engineers, for engineers

The SPECTRUM ACE™ 2a Starter Kit comes with the SPECTRUM ACE™ 2a CPU, all of the connectors shown, plus the SPECTRUM ACE™ Comm Board and a 5V DC 1 amp UL listed power supply.

$119.00

Our Advanced Control Environment (ACE) is composed of features that create an integrated solution for design, development, delivery and maintainance of even the most complex embedded systems.

ACE features ALEC™, our operating system with a built-in text editor plus integrated debugging and trace commands. This provides you with instant access to your code for debugging and optimization, whether on the bench or in the field.

Each SPECTRUM ACE™ Single Board Computer features I2I Controls' ALEC™ (Advanced Language for Embedded Control) operating system.

ALEC™ employs our Resident Run-Time Micro-Compilation process which produces highly compact code that typically executes between 30 and 40 thousand instructions per second.

The SPECTRUM LITE™ Evaluation Package comes with a SPECTRUM LITE™ CPU, the SPECTRUM LITE™ proto board featuring the 4 channel, 16-bit 1-wire DS2450, and a 5V DC 1 amp power supply.

$89.00

Edit your code in SRAM, then store your programs in the on-board FLASH or to an external USB drive. Since ALEC™ supports multiple boot modes, the user can opt to boot the system from SRAM, FLASH or USB. USB flash drives are supported with a suite of DOS-like disk instructions, providing storage for both bootable programs and/or run-time data. The USB interface board, along with many other add-on devices, can be ordered from our website.

www.i2icontrols.com

SUNSTONE CIRCUITS
THE EASIEST PCB COMPANY TO DO BUSINESS WITH

ValueProto™ PCBexpress® Full Feature

Sunstone Circuits® pioneered the online ordering of printed circuit boards and is the leading PCB solutions provider with more than 35 years of experience in delivering quality prototypes and engineering software. With this knowledge and experience, Sunstone is dedicated to improving the PCB prototyping process from quote to delivery (Q2D®).

Did You Know? Sunstone Offers:

- Controlled impedance testing
- Fine lines and spacing (.003)
- RoHS compliant finishes
- Free 25-point design review
- Free shipping & no NRE's
- Flex / Rigid Flex Boards
- Online Quote & Order
- PCB123® design software
- RF / Exotic Materials
- Over 99% on-time or early delivery
- Best PCBs in the industry
- Live customer support 24/7/365